ALGORITHMS AND HARDWARE TECHNOLOGY FOR IMAGE RECOGNITION

Final Report

to

U. S. Army Night Vision Laboratory
Fort Belvoir, VA 22060

under

Contract DAAG53-76C-0138
(DARPA Order 3206)

Prepared by

David L. Milgram
Azriel Rosenfeld
Computer Science Center
University of Maryland
College Park, MD 20742

Thomas Willett
Glenn Tisdale
Westinghouse Defense and Electronics System Center
Systems Development Division
Baltimore, MD 21203

March 31, 1978

ALGORITHMS AND HARDWARE TECHNOLOGY

FOR IMAGE RECOGNITION

Final Report

to

U. S. Army Night Vision Laboratory
Fort Belvoir, VA 22060

under

Contract DAAG53-76C-0138
(DARPA Order 3206)

Prepared by

David L. Milgram
Azriel Rosenfeld
Computer Science Center
University of Maryland
College Park, MD 20742

Thomas Willett
Glenn Tisdale
Westinghouse Defense and Electronics System Center
Systems Development Division
Baltimore, MD 21203

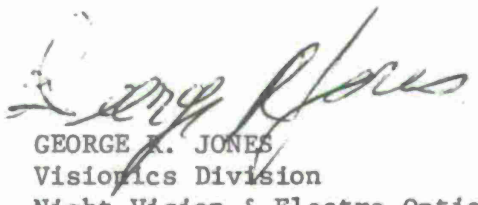
March 31, 1978

FOREWORD

The DARPA Smart Sensor Program, under the leadership of LTC Carlstrom, brought together a collection of teams each of which combined Educational and Industrial Laboratories. This report documents the contributions of the team consisting of the University of Maryland and Westinghouse Defense and Electronics System Center and the US Army Night Vision and Electro-Optics Laboratories.

The teamwork of Government, Educational Institution, and Industrial Laboratory was, in this case, highly productive. The initial concept design was influenced by an understanding of the limitations of hardware implementation and the hardware design considerations validly reflected the needs of system algorithm design. Both reflect well the constraints of militarily relevant missions.

The spirit of cooperation and productivity were enhanced by monthly program reviews attended by all three principal research personnel which allowed many iterations of the design process in a short period of time. The result was a great decrease in the delay usually encountered on transferring a new idea from basic research to implementation in military hardware.



GEORGE R. JONES
Visionics Division
Night Vision & Electro-Optics Laboratories

Table of Contents

1. Introduction

This is the Final Report on Contract DAAG53-76-C-0138, "Algorithms and Hardware Technology for Image Recognition", covering research conducted during the period 1 May 1976 through 31 January 1978. The Contract was funded under DARPA Order 3206, and was monitored by the U. S. Army Night Vision Laboratory, Fort Belvoir, VA. The project monitors were Mr. John S. Dehne and Dr. George R. Jones of NVL. The principal investigators were Profs. David L. Milgram and Azriel Rosenfeld of the University of Maryland. A subcontract, entitled "Recognition Technology for a Smart Sensor", was awarded to the Westinghouse Defense and Electronics Systems Center; it was directed by Dr. Glenn E. Tisdale, program manager, and Mr. Thomas J. Willett. Monthly meetings were held in which key NVL, Maryland, and Westinghouse personnel participated.

Under the contract, algorithms for detecting and classifying tactical targets on forward-looking infrared (FLIR) imagery were developed. The subcontract investigated the implementability of these algorithms in charge-coupled device (CCD) technology, and also successfully implemented one basic function, sorting. A list of the principal accomplishments under the contract is given in Section 2. Section 3 reviews the Maryland work on algorithms, including image modelling, pre- and post-processing, segmentation, feature extraction, and classification. Section 4 reviews the Westinghouse efforts.

A list of reports and publications generated under the contract is given in Section 5.

2. Summary of Accomplishments

1. Design and implementation of a comprehensive algorithm for object recognition in FLIR imagery with a detection rate above 95% and a false alarm rate between 1 and 2 false alarms per frame.

2. Fabrication and testing of a CCD sorter chip capable of operating at 3 megapixels/sec. The sorter function is a crucial step in several image operations including histogramming, median filtering, non-maximum suppression and connected component coloring.

3. Investigation of the cost, performance and constraint tradeoff in implementing a target cueing algorithm in CCD (charge-coupled device) technology. The resulting design is within specifications for usage in smart sensors.

4. Development of the "Superslice" algorithm for reliable region extraction based on the cooccurrence of border points of regions with locally maximum edge detector responses. This is an important example of the use of convergent evidence to strengthen assertions.

5. Design and analysis of statistical models for threshold selection, image operation response prediction, and optimal edge detection.

6. A new method for adaptive quantization of an image which reduces the number of gray levels present using only the histogram.

7. Comparison of image smoothing methods, including median filtering.

8. A study of shrink/expand noise cleaning schemes, including a local min/max method which cleans the image prior to thresholding.

9. Evaluation of a variety of edge detectors and the de-

velopment of a reliable method for edge thinning.

10. Construction of a "fuzzy" thinning algorithm which allows thinning to occur in gray level images prior to thresholding.

11. Development of methods for threshold selection based on gray level and gradient value.

12. Generalization of thresholding to the multiple object class environment with the ability to predict appropriate (gray level, gradient value) segmentation regions for the object classes present.

13. A variable thresholding scheme which produces a binary (or ternary) representation of an image.

14. An extension of threshold selection for sequences of images.

15. Simplification of the logic of the standard connected component coloring algorithm and its extension to produce a chain encoding of the component boundary in a single pass.

16. Implementation of Hyperslice: a recursive segmentation which improves the Ohlander region extraction method.

17. An algorithm for region tracking in image sequences using dynamic programming.

18. Comparison of features for target recognition.

19. Construction of a hierarchical classifier for target detection and recognition.

20. Development of Viewmaster - a software aid to assist in the construction of image processing programs.

The Westinghouse effort complemented the Maryland effort in several ways. Initially, assistance was provided to Maryland in describing the requirements for automatic cueing, and in developing a data base of FLIR imagery. A brief description of the function of Westinghouse cueing algorithms was offered for background information only.

During the conception and test of the Maryland cueing algorithms, Westinghouse carried out an investigation of techniques for their implementation, with particular emphasis on charge transfer devices. When processing functions were specified, a detailed analysis was then carried out so as to determine the feasibility of implementing them in CCD's. This process continued throughout the first year of the program.

During the final nine months, a specific circuit was chosen for the fabrication of a demonstration unit. A sorter function was selected because of its occurrence in several cueing operations. Chips were fabricated and tested at the Westinghouse Advanced Technology Laboratories, and a demonstration unit was assembled and shown at the Image Understanding Workshop in October, 1977. The unit rearranges a random series of pulses in ascending order by magnitude.

An estimate was also made of the area in monolithic silicon required to implement the cuer function in CCD's. The algorithm presently proposed by Maryland would require an area of 11-1/4 by 7-1/2 inches. If 3-inch by 3-inch modules were employed with 1/2-inch centers, an equivalent volume would be 3 inches by 3

inches by 6 inches. This volume has positive implications for missiles, i.e., Smart Missiles and other airborne platforms.

3. Algorithms

3.1 Introduction

This section describes the contribution made by the University of Maryland to the state of the art in FLIR target recognition and computer vision. The subsections describe the main areas of investigation and cover significant advances in image modelling and algorithm development. Not every idea investigated on the project was equally successful; this report reflects the main lines of effort, while other investigations are documented in the quarterly and semi-annual reports [1-4].

3.2 Image models

The work on image modeling conducted under this project was concentrated in three main areas:

1) Modeling of the joint (gray level, edge value) statistics of FLIR scenes, as a basis for defining threshold selection techniques.
2) Modeling of thresholding and edge detection responses to background regions, as a basis for predicting false alarm rates.
3) Modeling edges in images as a basis for defining optimal edge detection operations and for evaluating edge detector output.

This work is briefly summarized in the following subsections. References are given to earlier project reports in which detailed treatments can be found.

3.2.1 Model-based threshold selection

An approach to modeling FLIR imagery has been developed, based on the simplifying assumption that targets appear as homogeneous hot regions within a homogeneous cooler surround. This model describes the joint probability density of gray level and edge strength in such images, for various edge-detecting operations [1,2]. In brief, the model predicts that for low edge values (corresponding to points in the interiors of objects and background), there should be two relatively well separated probability peaks, of different sizes, representing the gray levels of object and background interiors, respectively. For higher edge values, corresponding to points on object/background borders, these peaks should merge together and become a single peak representing the border range of gray levels.

The model just described can be used as a guide to segmenting FLIR images by thresholding. At low edge values, it should be easy to pick a threshold at a gray level in the valley between the two probability peaks, since these are relatively well separated. At high edge values, the peak gray level value itself, or perhaps the mean gray level, should be a good threshold, since this represents the "center" of the edges. For intermediate edge values, one can compromise between these two thresholds in various ways. A comparative study of threshold selection schemes based on this approach has been conducted [5]. This work will be discussed further in Section 3.5.

3.2.2 Operator response prediction

a) Predicting results of thresholding

Thresholding images is a common process and much work has been directed towards selecting the correct value at which to threshold and estimating the expected error. Normally, one thresholds only images which contain some signal. Thresholding pure noise is to be avoided when possible. Since there may be occasions when thresholding noise is unavoidable (e.g., a poor threshold was chosen), it is important to predict the expected results. The expected number of above-threshold regions that result when noise is thresholded is useful in planning for data structure storage allocation and in predicting false alarm rates. When a bad threshold "breaks up" an object, knowledge of the expected sizes and shapes of noise regions can be used to help discriminate object fragments from noise. No methods currently exist for predicting the number of connected components of thresholded spatially correlated signal (or noise). However, it has been found possible [6] to estimate the moments of regions, the density of border points, and lower bounds on the number of connected components in thresholded noise images. The input grayscale image is modeled as a two-dimensional random process (stationary random field) characterized by its mean and power spectrum. Tests with both synthetic data (smoothed noise) and actual data were conducted to compare the predicted and measured responses.

The predictions are worst for thresholds at or near the mode of the noise distribution, but in general, the comparison showed reasonable agreement between the predicted and measured values.

b) Predicting edge detector response

Statistical response prediction for edge operators can be used to determine the nature of further processing of the response. If edge detector output is to be thinned or thresholded, the false alarm and false dismissal rates depend on the statistics of the operator responses. A study has been conducted [7] which discusses the statistical properties of the outputs of some edge detectors operating on a general class of images.

The image model considered is the same as in (a) just above; this model is appropriate for predicting the response of edge detectors to background noise. The edge detectors analyzed are the Laplacian and its absolute value, and the absolute difference of averages over adjacent 2x2 and 4x4 neighborhoods. The response features which were measured are the mean edge response at each point, the variance, the auto-covariance, and the cross-covariance of gray level and edge value at a number of displacements. In addition, the density of the local maxima of edge values was computed. Tests using a set of synthetic background images showed good conformity to the predicted features.

3.2.3 Edge modeling

a) Optimal edge detection

Many optimality criteria have been proposed for edge detection. Among the most well known is that devised by Hueckel [8,9]. It involves fitting an ideal parameterized step edge to the image data so as to minimize the mean squared error. A new optimal detector has been designed [10] that simplifies several assumptions associated with the Hueckel detector and thereby solves an easier optimization problem. Specifically, by assuming that the local mean is zero and the local variance is unity, two Hueckel parameters can be eliminated. Further simplifications follow if the operator can be applied at each point with the edge assumed to pass through the center (or not to exist at all). The resulting formulation can be tuned to favor edges with known a priori probabilities. The computational effort involved in applying the operator may be reduced by solving the associated cubic equation using a simple iterative approximation, such as Newton's formula. Testing on actual data has verified that this approach provides greater sensitivity to edge orientation than a previously proposed [11] simplification of the Hueckel operator.

b) Evaluation of edge operators

The Hueckel operator defined in [9] has been found to incorporate a theoretical flaw leading to eccentric behavior in textured images. An operator which is conceptually

similar but apparently more dependable has been defined [12]. Comparative tests have been made of this and several other operators (including Hueckel's [9], its simplification [11], and the new optimal operator defined in [10], as well as the very simple Sobel operator), to evaluate their adequacy in obtaining the magnitude, direction, and reliability of the edge response at some set of image points, for both ideal and distorted images. The performances of these operators were, in general, closely related to their sizes (and hence, to their computational costs). All of the local operators were able to detect the directions of distorted edges on small (6x6) domains to an accuracy of about 10°, and their magnitudes to within about 10%. On larger (9x9) domains, angular resolution was improved, but ramps became significant as a source of spurious responses. The Sobel operator was judged to perform better than the operator of [11]. The operator of [10] was better able to reject ramps on larger domains, but it is more expensive to apply than the other local operators. The regional operators of [9] and [12] performed similarly; the latter was less affected by the presence of imperfections.

3.3 Preprocessing

Preprocessing refers to those transformations applied to the raw image data for the purpose of correcting, simplifying and regularizing the imagery. The resulting images should therefore be more amenable to further processing and more alike in certain properties essential to subsequent algorithms. Thus, for example, sampling and windowing reduce the size of the image to be processed. Histogram transformation and requantization convert all image quantization levels to a range which facilitates feature extraction. Smoothing reinforces regional uniformity and decreases the effects of certain kinds of noise.

Preprocessing steps are best justified by the problem environment itself. A knowledge of the sensor characteristics and geometry will suggest various kinds of radiometric and geometric corrections. For example, with FLIR data, the image is best understood as an array of thermal measurements. If these measurements reflect the ground truth, then much more subtle distinctions can be made; recognizing that a particular temperature is beyond the normal range for a vehicle can, perhaps, indicate that the vehicle is on fire. Similarly, a range map converting pixels to actual size/area measurements can allow a viewer or a program to gauge the size of a particular region and thereby discern its identity more reliably.

In the problem environment at hand it was not possible to acquire substantial information concerning the sensor or the imaging situation, due to classification problems. Thus the "intelligent" corrections of the previous paragraph were impossible. However, several preprocessing steps do make sense. The following subsections describe them.

3.3.1 Sampling

According to the sampling theorem, the spacing of the data points should be half the size of the smallest feature to be detected. Thus, to detect objects of one meter on a side, pixels should correspond to one half meter on a side. In practice, however, the presence of noise demands that the data be redundant to increase the reliability of the extraction process. A finer resolution can often provide this redundancy. Naturally, the price must be paid in additional processing time. This tradeoff is difficult to model analytically especially since many different features are extracted from an image and their relative importance is difficult to assess.

Two processes for region extraction are paramount for our work -- thresholding for whole region extraction and edge detection for region border verification. Of these two, edge detection is more sensitive to noise. The degree of sampling allowable for the image data set should therefore not be so great that reliable edge extraction is compromised. A 2-to-1 size reduction (eliminating every other row and column) was found to be compatible with reliable edge detection. In the unsampled images, the average edge ramp cross-section was found to be about 5 pixels wide; thus a 2-to-1 reduction gave about a 3-pixel edge ramp which was consistent with the need to localize edges fairly accurately.

An alternative approach attempted to reduce high frequency noise by extracting windows based on 2x2 averaging rather than sampling. Thus, instead of discarding every other row and column, each pixel in the sampled image was the average over a (disjoint) 2x2 neighborhood in the original image. A smooth, less noisy image was produced and row dropouts were partially eliminated. However, the images seemed to have less contrast. Sampling followed by smoothing appears to be better than smoothing followed by sampling.

A major emphasis in the project has been the detection of small or faint targets. For this reason, the sampled images were also windowed so as to capture the target regions and to further reduce the computational load. Naturally, one must avoid techniques which assume that each window contains exactly one target in its central region. This dilemma asserts itself in subtle ways. Statistical properties of the window, e.g., histogram, central moments, etc. are good predictors of object presence, threshold, etc. However, they cannot be employed in practice unless window size is correctly estimated and window border situations are handled. Our approach does not depend on window size (or frame size) and therefore windowing is an appropriate preprocessing step. Note, however, that estimates of the false alarm rate cannot be reliably derived solely from target windows. For this reason, small noise windows (containing no targets) and large windows (consisting mainly of background clutter) were also processed.

3.3.2 Histogram transformations and adaptive quantization

Sensor output is related to actual phenomena according to physical laws. If this correspondence is well understood beforehand, it is possible to correct and transform the data to improve subsequent processing. Thus if FLIR data could be used to estimate reliably the temperature of objects, then quite stringent tests could be made to enhance recognition rates. Unfortunately, the analytic interpretation of FLIR data at long range is complicated by many effects such as sun-angle, wind, smoke, surface composition, etc. Furthermore, the sensor hardware itself is subject to unpredictable electronic noise, disturbances and failures. Only some of these effects can be alleviated and then (due in part to the classified nature of the sensor) only statistically.

Among the conventional gray level modifications considered useful for producing more manageable imagery are the rather simple histogram mapping techniques. Figure 3.3.1a illustrates the gray level histogram of an unmodified image. The gray level range is defined by eight bits-256 gray levels--and can be seen to exhibit significant non-uniformities of response. Moreover, from a processing point of view, 256 gray levels do not effectively reflect the true gray level range and contrast. A simple 2-bit shift operation, converting 8-bit pixels to 6 bits, has the effect seen in Figure 3.3.1b of smoothing the histogram while reducing the gray level

Figure 3.3.1a. 256-level histogram.

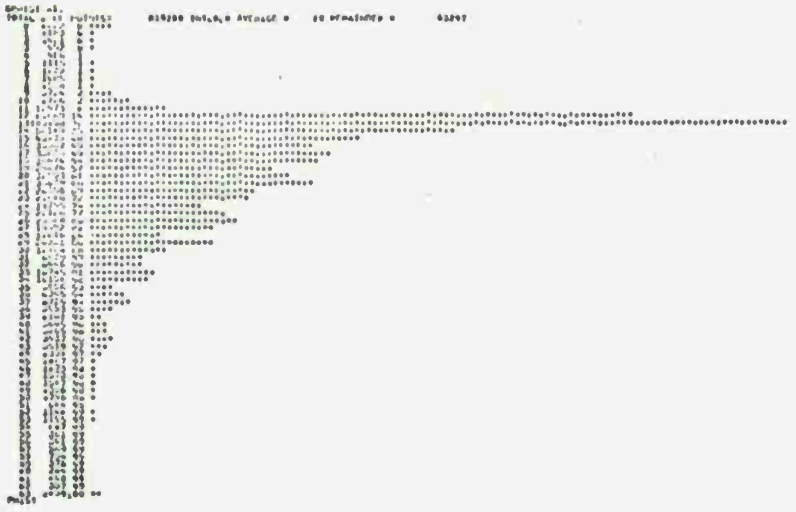
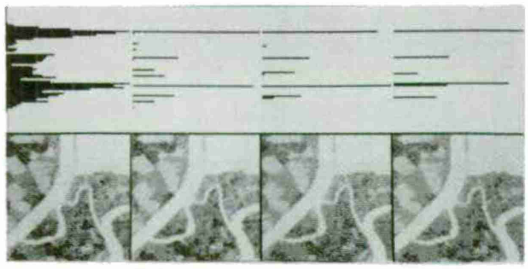


Figure 3.3.1b. Same histogram re-quantized to 64 levels.

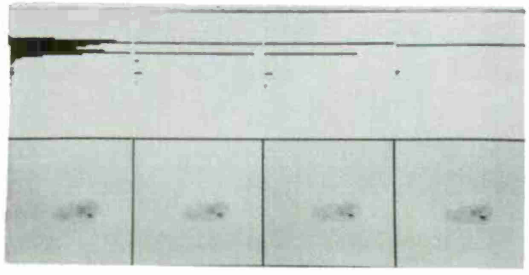
range to 64 gray levels. This technique if continued for further shifts would ultimately combine significant peaks corresponding to object/background contrast. However, the conversion from 8-bit to 6-bit was found to be justified, as it alleviated non-uniform sensor response without destroying target discriminability.

If one assumes that a scene consists of the juxtaposition of objects of uniform temperature taken from a small number of such temperatures, then it is possible to convert the image into one with only a few different gray levels present. An attempt at adaptive requantization is described in [13]. Briefly, an iterative process constructs a new histogram from the previous version by identifying gray level peaks and having them gain strength (i.e., points) from neighboring non-peaks while the non-peak areas are thereby depleted. The result is a mapping from the original gray level domain to a new sparse set of gray levels. The resulting requantized images (Figure 3.3.2) seem not to have lost object/background discriminability.



a.

b.



c.

Figure 3.3.2. Result of four iterations of the peak sharpening process using neighborhood sizes of 2, 3, and 4 for (a-b), sizes 2, 4, and 5 for (c).

3.3.3 Image smoothing

In the previous section, preprocessing steps were described which contributed to the interpretation of a scene as a mosaic of uniform sensor responses. The techniques considered the gray level population only. Proximity was not involved. In this section, we discuss attempts to smooth the image spatially so that nearby points from the same region will have more nearly identical gray levels. There are a number of justifications for spatial image smoothing. First, by making the image spatially more uniform, one increases the probability that points belonging to the same region will be treated identically. Thus, the point sets extracted by thresholding will appear better defined with fewer pinholes and fewer isolated points. This is reasonable since the chosen image resolution is intended to cover any object with numbers of pixels. The second reason for smoothing is to eliminate insignificant local changes of contrast. Otherwise the output of edge detection operations based on differencing would contain many tiny spurious edges which tend to obscure the proper edge signals. Third, the statistical properties of a smoothed image are more representative of the true situation. Thus, many decisions based on the statistics of the smoothed image are more reliable.

Two methods of image smoothing were investigated. In a first attempt, the mean value of a fixed neighborhood about each point replaced the point's value. Figure 3.3.3a

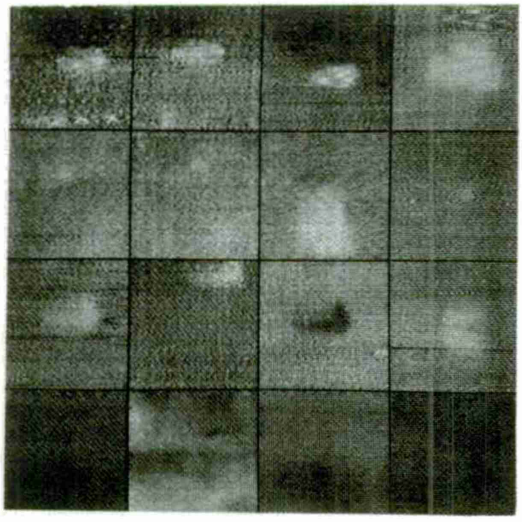
shows the effect of replacing each point of a step edge by its mean value (blurring). As is evident, blurring smears edges. Figure 3.3.4a-d illustrates blurring for several target windows, and also shows the histograms of these windows before and after blurring. Note that blurring tends to blend peaks in histograms, thus making thresholding more difficult. Also, small faint objects tend to become less distinct.

A second approach to image smoothing has the property of preserving edges. At each point of an image, the median value of the gray levels over a kxk neighborhood is computed. The value of k depends on the amount of local noise variation. For the original images, a 5x5 neighborhood size was chosen. Figure 3.3.3b illustrates the effects of median filtering on a step edge. Note that the median does not increase the ramp width. Thus edges do not smear. This is demonstrated in the two-dimensional case in Figures 3.3.5-3.3.7 for a tank image. Median filtering does, however, round off sharp corners. This was not a serious problem in this data base. Figure 3.3.4e-f illustrates a number of median filtered windows and their histograms. The general algorithm for median computation over k^2 points is of order k^2. However, better results may be obtained when evaluating a running median, by making use of the high autocorrelation of gray level in most images. The cumulative histogram of the k^2 data points is maintained in a vector of length d (e.g., d=64). The k deletions and k insertions are interleaved in pairs.

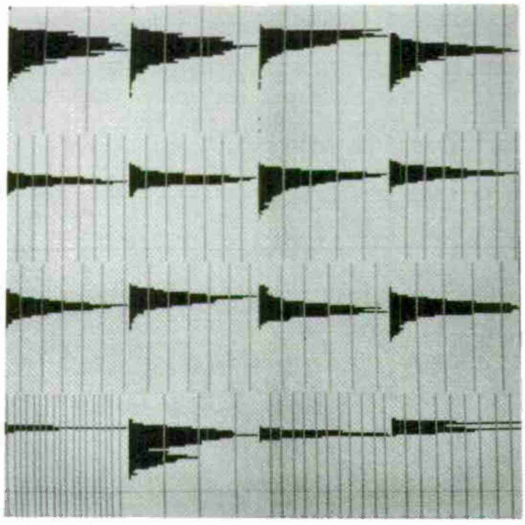
a. Mean filtering

b. Median filtering

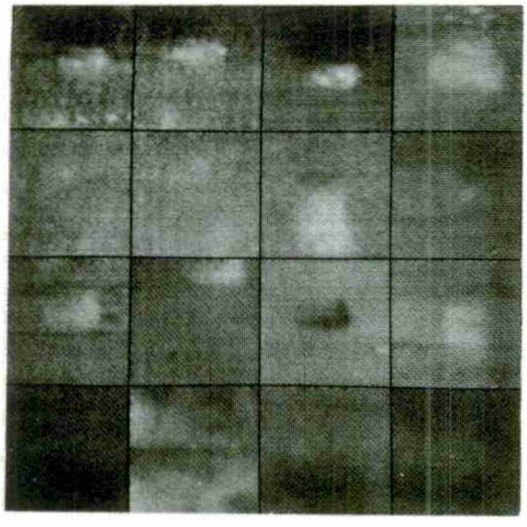
Figure 3.3.3. Effect of filtering on step edges using a five point neighborhood.



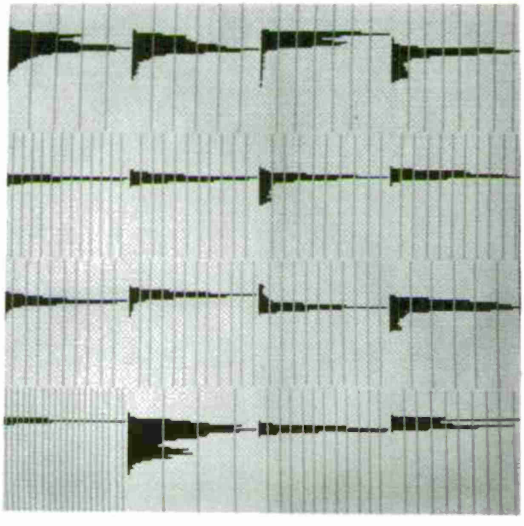
a. Originals.



b. Histograms of (a).



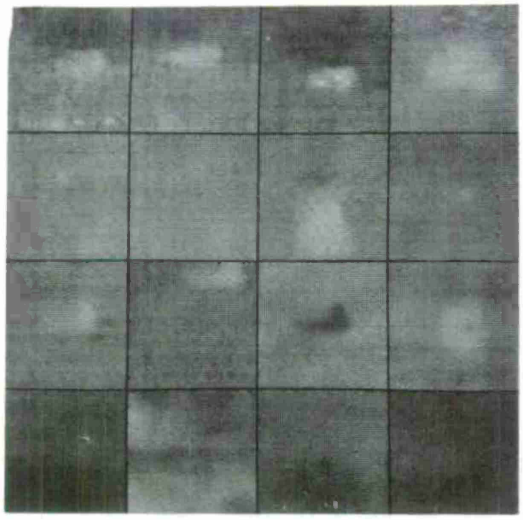
c. 3x3 mean filtered windows.



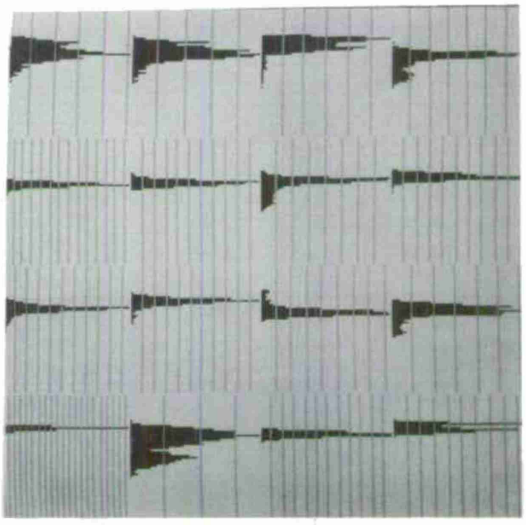
d. Histograms of (c).

Image Reference:				
	3R	4T	6T	24T
	34R	35R	41R	52R
	21A	22A	23A	37A
	14N	20N	26N	38N

Figure 3.3.4. Comparison of mean and median filtering.

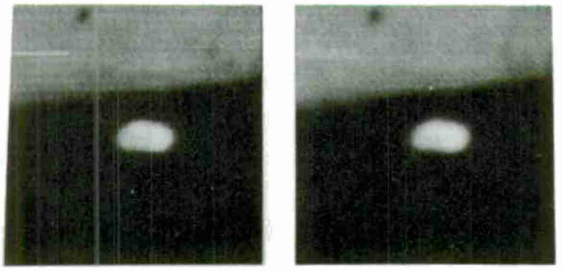


e. 3x3 median filtered windows.

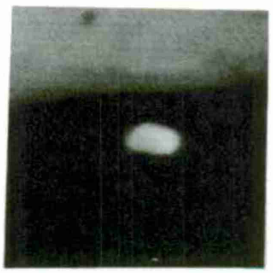


f. Histograms of (e).

Figure 3.3.4 (continued)



a. b.



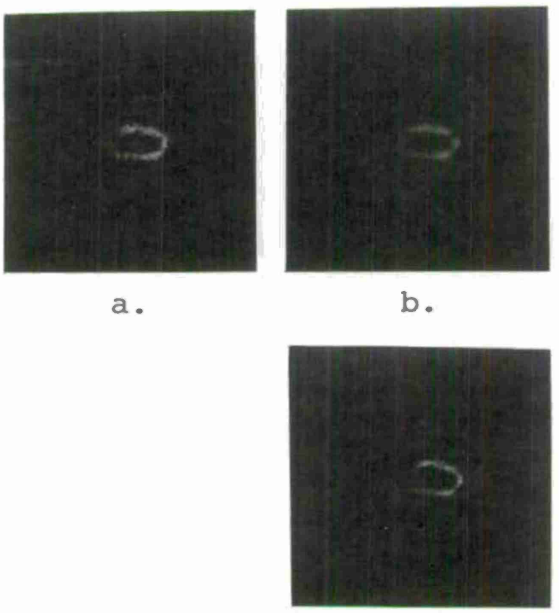
c.

Figure 3.3.5. Gray level images.

a. Original FLIR image of a tank. Note the noise content and the presence of a thin noise line at the upper left.

b. Mean filtered image using a 5x5 square neighborhood at each point. The tank appears blurred, as does the border between the road (dark) and the grass (light). The thin noise line is smeared into the background.

c. Median filtered image using a 5x5 square neighborhood. The tank contours appear sharper, while overall the image has been smoothed.



a. b. c.

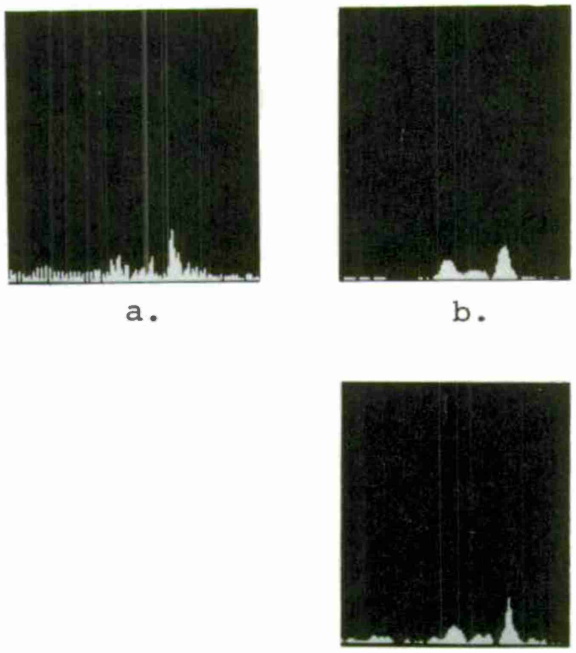
Figure 3.3.6. Results of Edge Detection

Each of the windows was subjected to an edge detection operation which detects the most significant edge at each point over four orientations. Note that edges surround the various regions in the image but that the edges in the median filtered image (c) are sharper and have more contrast than those in the mean filtered image (b).

a. Edge detection response for the original image.

b. Same as above for the mean filtered image.

c. Same as above for the median filtered image.



a. b. c.

Figure 3.3.7. Edge Cross Sections

A single line of the edge detection image passing through the tank is displayed with height corresponding to edge value. Notice that the median filtered response exhibits a thinner, higher peak corresponding to a sharper, more contrasting edge than the mean filtered response.

a. A single line of edge response from Figure 3.3.6a.

b. Same as above for Figure 3.3.6b.

c. Same as above for Figure 3.3.6c.

Each (deletion, insertion) pair isolates a region of the vector which must be modified. The smaller this region on the average, the less work to be done. If the deletion and insertion in a given pair affect the same bin, no change is necessary. The length of the region of change in the cumulative histogram is the expected gray level difference of points at distance k. This corresponds to a variogram value, v(k). After updating, the vector is binary-searched for the median. Thus the number of vector operations is $k \cdot v(k)$, followed by log d operations to binary-search the updated vector. The sum $k \cdot v(k)$ + log d should be quite small for relatively smooth images.

3.4 Edge detection

The extraction of edge features has proved useful in a number of project areas. Section 3.5 will describe threshold selection methods which utilize edge information. Edges are used also in the critical step of the Superslice algorithm (Section 3.7). In this section, we discuss the variety of edge operations investigated and a method for thinning edge response so as to locate the apparent edge.

3.4.1 Comparison of methods

Methods for edge detection abound in the literature (for a survey, see [14]). Some of the simplest methods involve convolutions of templates with an image. A number of these were considered in the current work. These include:

Laplacian: $|e - (a+b+c+d+f+g+h+i)/8|$, where the neighborhood of e is

```
a b c
d e f t
g h i u
  v w
```

Roberts Gradient: $\max\{|a-e|, |b-d|\}$.

Three-by-three: $\max\{|a+b+c-g-h-i|\ ,\ |a+d+g-c-f-i|\}$

2x2 Difference:

$1/4*\max\{|d+e+g+h-f-t-i-u|, |b+c+e+f-h-i-v-w|\}$.

(In other words, the value corresponds to the maximum of the differences between 2x2 averages over adjacent pairs of horizontal and vertical neighborhoods. This scheme extends to diagonals also.

4x4 Difference: This is the same as the 2x2 difference except that averages are taken over 4x4 neighborhoods.

8x8 Difference: The same as the previous except that averages are taken over 8x8 neighborhoods.

Experiments with these operators indicated that the Laplacian (which is a second difference), the Roberts Gradient and the 3x3 Gradient were too sensitive to minute changes in

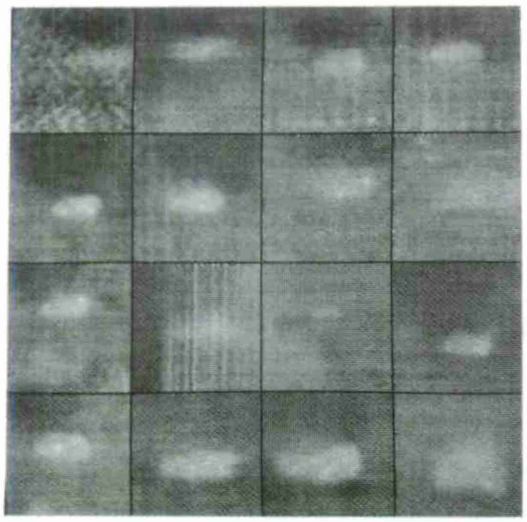
gray level. The differences of averages operators produced better output by virtue of the increased amount of smoothing on each side of the edge. Knowledge that typical edge width in the windows was three pixels suggested that the 4x4 operator could span the edge ramp (to give the maximum gradient value) while remaining sensitive enough to detect the edges of small faint regions.

3.4.2 Edge thinning

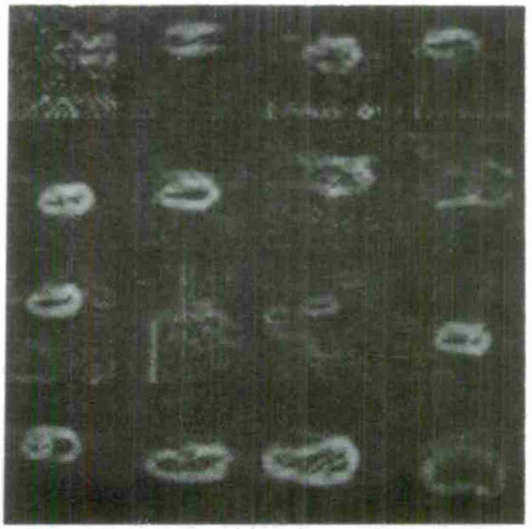
In the world of man-made objects, edges correspond to the juxtaposition of surfaces or shadows. In a well focused image, edges should appear sharp and should extend in some direction for some length. (In the natural world, the boundaries of regions are not necessarily as sharply defined, e.g., for trees, fields, etc.) The output of the operators described in the previous section, however, are generally smeared at or near the true edge location. Nonetheless, for certain types of image understanding it is necessary to localize the edge so that it lies along the object boundaries. Given a knowledge of the edge detector, it is possible to design a process which accepts the output of the operator and which produces a thinned representation of the edge at the location of maximum edge response.

It is not sufficient to consider simply those points of maximum response, since this would force adjacent points in the direction of the edge to compete. This problem can be alleviated by taking into account the computed local direction of the edge and by placing into competition only those points which are normal to the direction of the edge. In practice, a directional mask is associated with each edge point oriented normal to the direction of the maximum edge response at that point. The center point is then deleted (assigned zero response) if any point within the mask has a greater response. There are four masks:

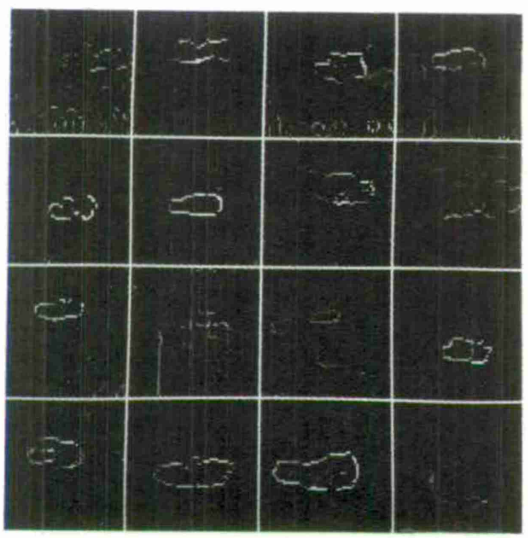
one associated with each principal edge direction. The process, called "non-maximum suppression," operates simultaneously on all edge values to produce a "thinned" edge map. Figure 3.4.1 illustrates the process.



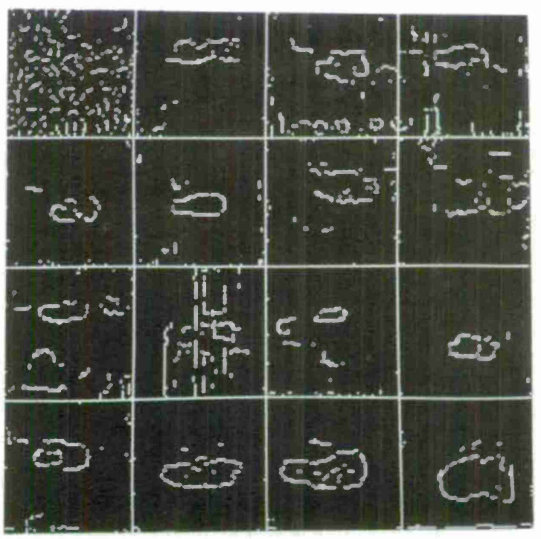
Originals.



Edge detector output.

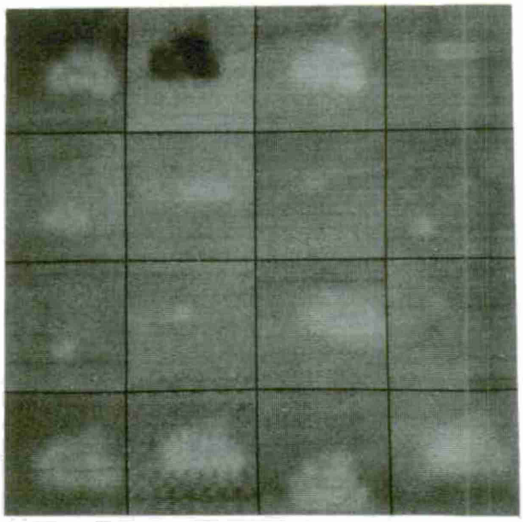


Thinned edge map.

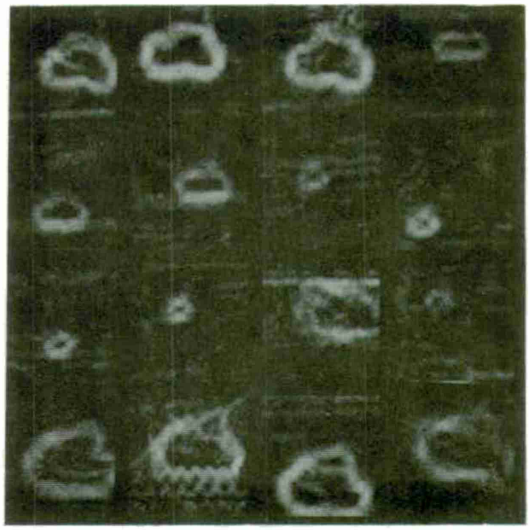


Thinned edge map thresholded to display only edge values > 2.

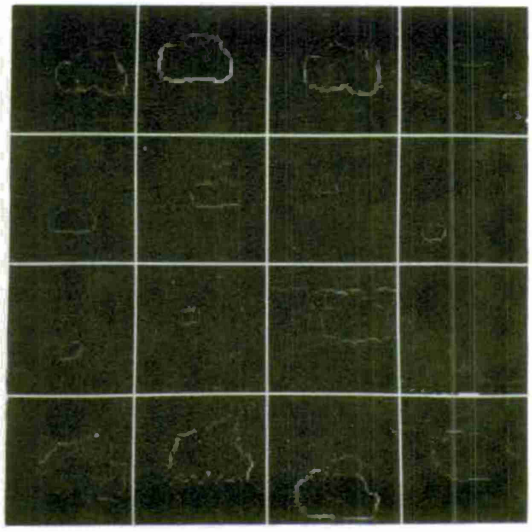
Figure 3.4.1. Results of edge detection and non-maximum suppression on 43 tank windows.



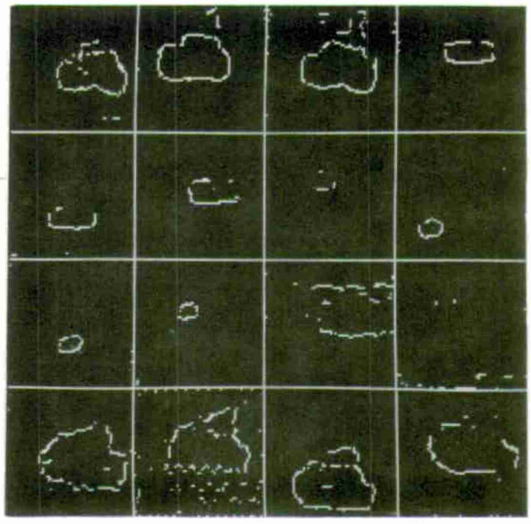
Originals.



Edge detector output.

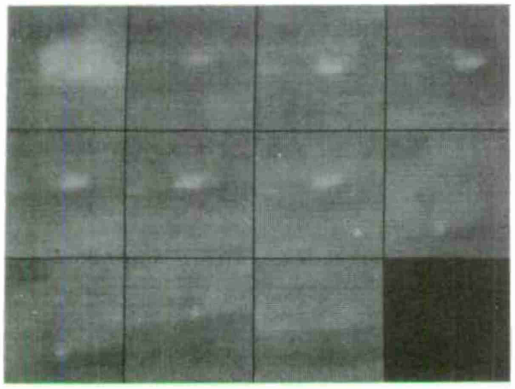


Thinned edge map.

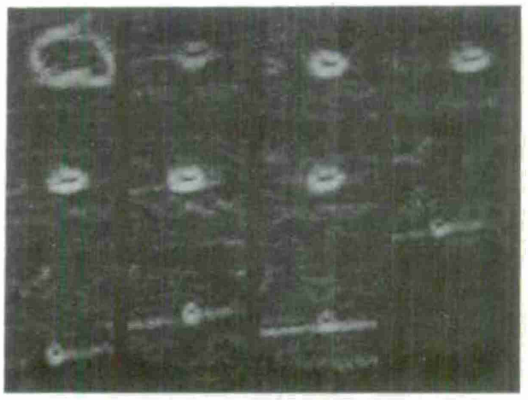


Thinned edge map thresholded to display only edge values > 2.

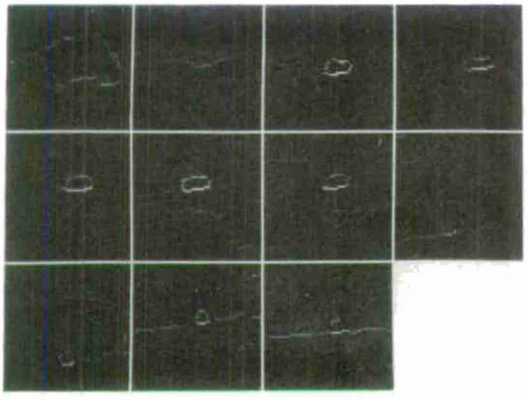
Figure 3.4.1 (continued)



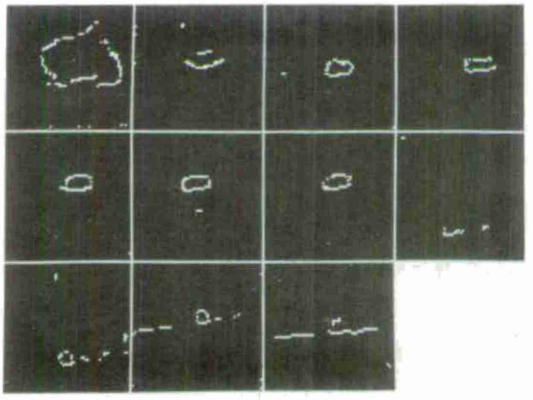
Originals.



Edge detector output.



Thinned edge map.



Thinned edge map thresholded to display only edge values > 2.

Figure 3.4.1 (continued)

3.5 Threshold selection

The properties of a pixel in a single sensor image are its position and its gray level. Our knowledge of the imaging environment allows us to predict an object's gray level more accurately than its position. In fact, the whole point of cueing is to locate a target. Thus one has little a priori information about target position; however, inasmuch as gray level is related to thermal emission in FLIR data, there are some fairly powerful heuristics available to aid in target recognition. For example, we may choose to assume that operating vehicles are warmer than the immediate background and that they radiate uniformly over their surfaces. Naturally, such assumptions are not always possible. In cold weather, metal loses heat faster than the ground; at close range, fine thermal detail is visible and the uniformity assumption fails. Nor are these assumptions meant to be exclusive, e.g., we do not claim that every object region warmer than its surround is a target. The power of these heuristics is to suggest approaches which capture essential problem domain knowledge.

In this project, the force of the heuristics of the previous paragraph is to emphasize methods which isolate distinct gray level regions from their surrounds. The simplest of such methods is thresholding -- the assignment of all points whose gray levels are greater than a predetermined level (the threshold) into a single class of

potential object points. The filtering, aggregation and ultimate classification of these points are the subjects of subsequent sections. In this section, we discuss our investigations of numerous methods for single and multiple threshold selection.

There is a progression in these methods which corresponds on the one hand to the need for increased sensitivity in choosing the "right" threshold and, on the other hand, to the deemphasis of the committment to that particular threshold. However, we still retain the notion that for each object region in the image there is a "best" threshold. In the worst case, every possible threshold yields a target region which is invisible (unextractable) at every other threshold. One must therefore be prepared to threshold at any gray level and within that thresholded image to discern the target regions and to ignore the noise regions. These last comments appear to call for the selection of every gray level as a value at which to threshold. Indeed, given sufficient parallel hardware and powerful target/noise discrimination criteria, this brute force approach could lead to a reliable and sensitive target cuer. A further discussion of this option is in Section 3.7 and some relevant experiments are to be found in Section 3.9. The remainder of this section describes techniques for finding appropriate thresholds when hardware and throughput considerations allow only a few thresholds to be utilized per frame.

3.5.1 Threshold selection based on edge values

In Section 3.2.1 a model was proposed for images consisting of objects and background, each with characteristic gray level distributions. If the gray level histogram of the image is markedly bimodal, one may choose the threshold at the valley between the two peaks (possibly shifted towards the smaller peak when using a maximum likelihood estimate). However, the smaller the object, the less likely the histogram is to exhibit strong bimodality. The background distribution engulfs the object's gray level range and tests for bimodality are inconclusive.

One approach [15] to solving this problem has been to select from the original image a set of points that are as likely to fall within the object as within the background. If one examines the output of operators which respond to edges, then high values should correspond to points falling at or near object edges. The mathematical model has shown the gray level distribution to be unimodal with a peak at the mean. Thus, these points are as likely to lie on the object as on the background and their mean value should correspond to the desired threshold.

A brief description of the threshold selection method is as follows: Let $e(i,j)$ be the edge value computed at (i,j), and let $g(i,j)$ be its gray value. Then the chosen threshold is $\bar{g} = AVG\{g(i,j) | e(i,j) \geq t\}$, where t is lowest edge value considered significant. Computationally, two arrays are

needed. One array, $TOTAL_0,\ldots,TOTAL_{63}$, accumulates the gray level g for each edge value e between 0 and 63; i.e., $TOTAL_e = TOTAL_e + g$. The second array, $N_0,\ldots,N_{63}$, tallies the number of points at each edge value. The desired average gray level $\bar{g} = (\sum_{i \geq t} TOTAL_i)/(\sum_{i \geq t} N_i)$.

Two parameters were treated in the experimental work: the choice of edge operator and the edge significance level t. Previous work with edge operators indicated that the 4x4 difference of averages operator was superior to the others as an edge detector in FLIR scenes. Experiments in threshold selection showed that thresholds chosen based on this operator were better overall [1].

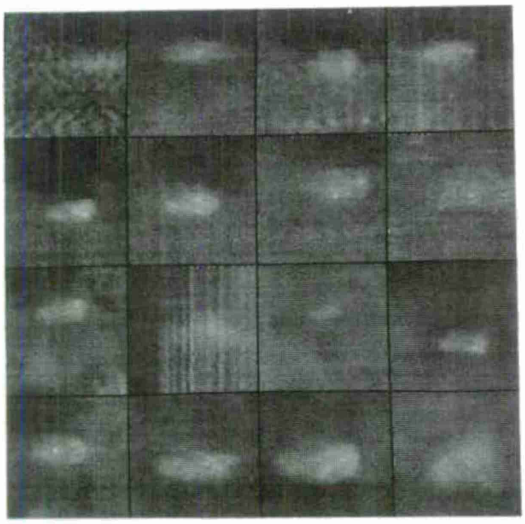
The proper selection of a value for t is important because this parameter controls the size and quality of the sample points of high edge value used to compute the gray level threshold. Setting t too high decreases the statistical reliability of the sample; while a small t may admit too many noise values. The choice of t depends on the expected amount of object edge. Obviously, many assumptions are built into this notion, e.g., that the window contains only a single object of known size, shape, contrast, resolution, etc. In a tactical situation, one could make estimates of these parameters based on situation data. Based on estimates of target size, t was chosen as the edge value corresponding to the 95^{th} percentile. This estimate was shown by experiment

to provide good thresholds for the windowed data set. This is illustrated in Figure 3.5.1.

The sensitivity of the chosen gray level threshold to different choices of t was tested and a graph of the threshold was plotted as a function of the gradient cutoff t; see Figure 3.5.2. There is a tendency for this graph to drift toward the mean gray value as t is decreased. The chosen threshold is stable for large objects. For small objects, the choice is quite sensitive to the bin size.

The approach above and several variations [5] can viewed as methods of decision surface selection in (gray level, edge value) space. This space is visualized as a two dimensional histogram with gray level along one axis and edge value along the other. Figure 3.5.3 displays such a 2-D histogram for a hypothetical object on a background. Points at A represent background while object points (perhaps with some noise) cluster at B. The bottom part of the U-shaped region contains high-edge value points. As we have pointed out, the average gray level of these points is a good threshold. Figure 3.5.1 illustrated 2-D histograms for several target windows.

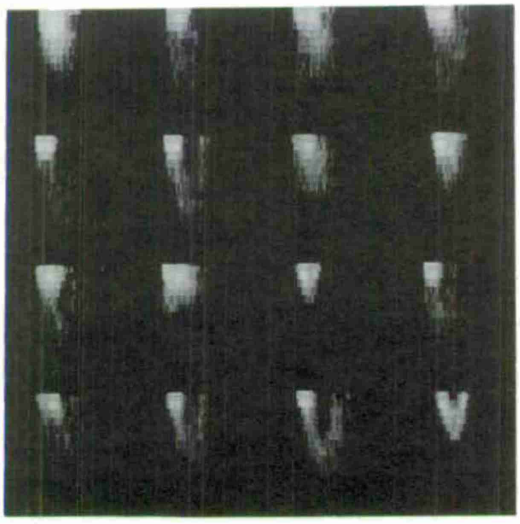
One may consider a threshold as a vertical decision surface separating object from background. Non-vertical partitions of the space have also been investigated [2] and were found to be capable of adding more points to the boundaries of object regions without substantially increasing



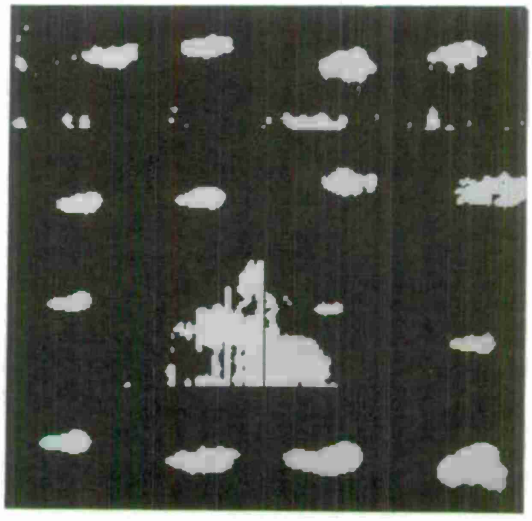
Originals.

1T	2T	3T	4T
6T	8T	9T	10T
11T	12T	13T	14T
15T	16T	17T	21T

Image reference numbers.

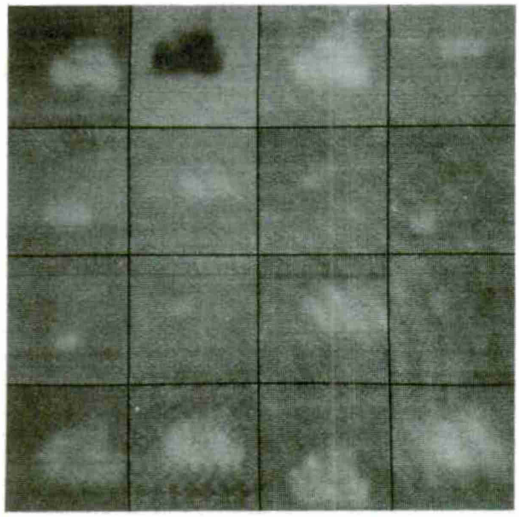


2-D Histograms.



Thresholded windows after shrink/expand (see Section 3.6.1).

Figure 3.5.1. Results of thresholding and post-processing 43 tank windows.

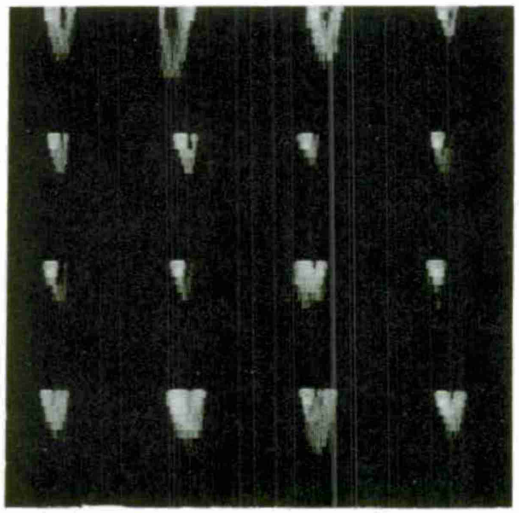


22T 23T 24T 26T

28T 31T 32T 33T

34T 35T 38T 40T

42T 43T 45T 46T



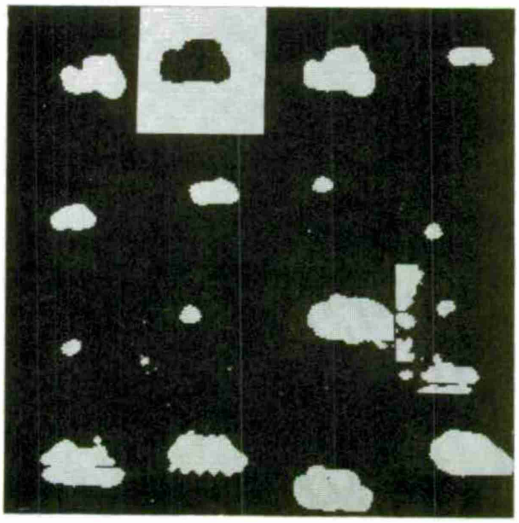
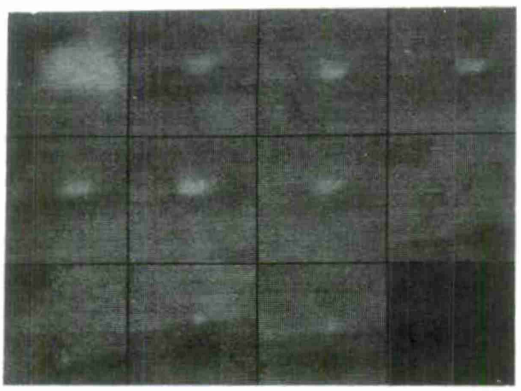


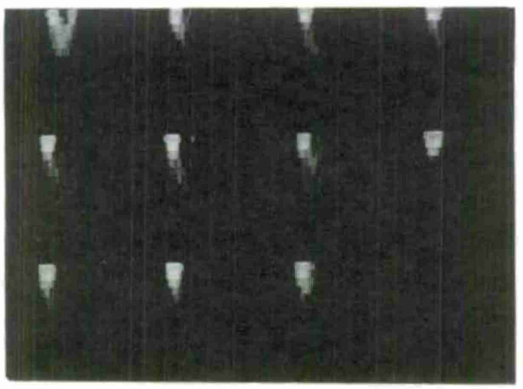
Figure 3.5.1 (continued)



48T 50T 51T 52T

53T 54T 55T 56T

57T 58T 59T



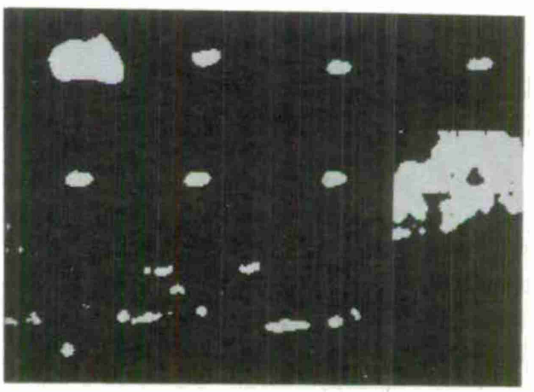


Figure 3.5.1 (continued)

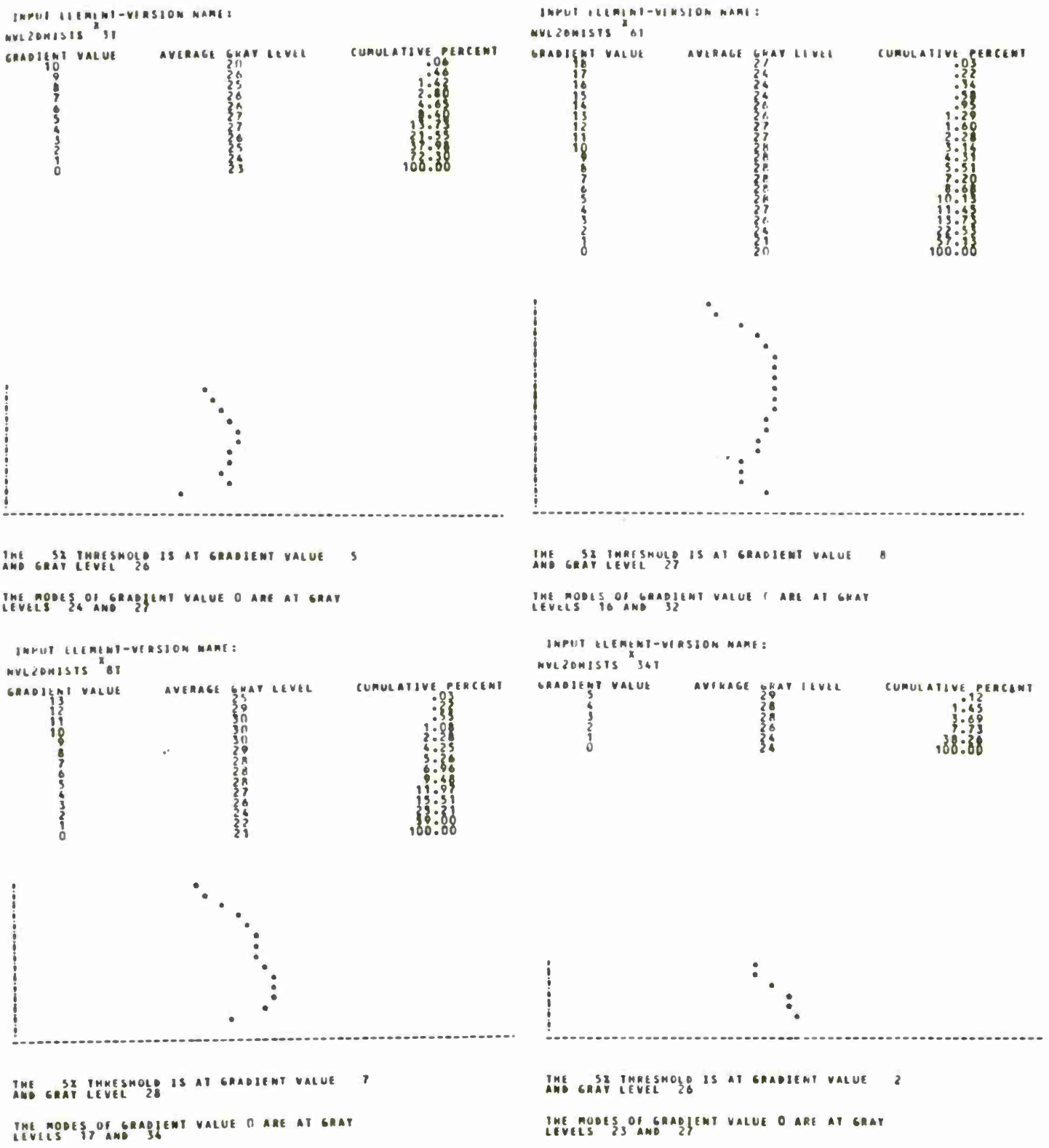


Figure 3.5.2. Graph of selected threshold as a function of percentage edge value cutoff. Abscissa: gray value increases to the right; ordinate: gradient decreases up the axis.

INPUT ELEMENT-VERSION NAME:
NVL2DHISTS 54T

GRADIENT VALUE	AVERAGE GRAY LEVEL	CUMULATIVE PERCENT
9	29	.15
8	26	.40
7	28	.77
6	26	1.43
5	26	2.40
4	25	3.82
3	24	5.70
2	24	10.90
1	21	35.58
0	21	100.00

THE 5% THRESHOLD IS AT GRADIENT VALUE 5
AND GRAY LEVEL 24

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS 22 AND 30

INPUT ELEMENT-VERSION NAME:
NVL2DHISTS 58T

GRADIENT VALUE	AVERAGE GRAY LEVEL	CUMULATIVE PERCENT
6	24	.12
5	23	.14
4	23	1.14
3	22	2.74
2	21	8.19
1	21	37.68
0	20	100.00

THE 5% THRESHOLD IS AT GRADIENT VALUE 2
AND GRAY LEVEL 21

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS 20 AND 22

INPUT ELEMENT-VERSION NAME:
NVL2DHISTS 51R

GRADIENT VALUE	AVERAGE GRAY LEVEL	CUMULATIVE PERCENT
4	29	.74
3	28	2.43
2	28	5.23
1	26	28.33
0	26	100.00

THE 5% THRESHOLD IS AT GRADIENT VALUE 2
AND GRAY LEVEL 28

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS 25 AND 29

INPUT ELEMENT-VERSION NAME:
NVL2DHISTS 52R

GRADIENT VALUE	AVERAGE GRAY LEVEL	CUMULATIVE PERCENT
5	23	.03
4	23	.58
3	23	1.54
2	24	3.14
1	23	63.64
0	22	100.00

THE 5% THRESHOLD IS AT GRADIENT VALUE 1
AND GRAY LEVEL 22

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS 21 AND 23

Figure 3.5.2 (continued)

```
 INPUT ELEMENT-VERSION NAME:
NVL2DHISTS  S6A

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      6                    26                      .25
      5                    26                      .88
      4                    26                     1.88
      3                    24                     6.06
      2                    24                    21.97
      1                    24                    58.39
      0                    23                   100.00

THE  50% THRESHOLD IS AT GRADIENT VALUE   3
AND GRAY LEVEL  24

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS  22 AND  25
```

```
 INPUT ELEMENT-VERSION NAME:
NVL2DHISTS  S4A

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      3                    27                      .68
      2                    27                     5.57
      1                    25                    22.75
      0                    25                   100.00

THE  50% THRESHOLD IS AT GRADIENT VALUE   1
AND GRAY LEVEL  25

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS  24 AND  26
```

```
 INPUT ELEMENT-VERSION NAME:
NVL2DHISTS  S1A

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      6                    27                      .25
      5                    27                      .77
      4                    24                     1.56
      3                    25                     3.32
      2                    24                    10.68
      1                    23                    47.71
      0                    23                   100.00

THE  50% THRESHOLD IS AT GRADIENT VALUE   2
AND GRAY LEVEL  24

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS  22 AND  25
```

```
 INPUT ELEMENT-VERSION NAME:
NVL2DHISTS  S5A

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      7                    25                      .36
      6                    27                      .98
      5                    27                     2.40
      4                    24                     4.31
      3                    24                     5.69
      2                    25                     9.60
      1                    24                    61.21
      0                    25                   100.00

THE  50% THRESHOLD IS AT GRADIENT VALUE   3
AND GRAY LEVEL  25

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS  22 AND  26
```

Figure 3.5.2 (continued)

```
 INPUT ELEMENT-VERSION NAME:
                X
NVL2DHISTS  2N

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      6                    29                      .03
      5                    30                      .18
      4                    31                      .95
      3                    30                     3.57
      2                    29                    15.05
      1                    28                    57.86
      0                    28                   100.00

!
!                          .
!                           .
!                            .
!                             .
!                            .
!                           .
!
-------------------------------------------------------------


THE   5% THRESHOLD IS AT GRADIENT VALUE    2
AND GRAY LEVEL   28

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS   27 AND   30


 INPUT ELEMENT-VERSION NAME:
                X
NVL2DHISTS  2GN

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      5                    23                      .06
      2                    22                     4.56
      1                    22                    46.54
      0                    22                   100.00


!
!                      .
!                      .
!                       .
!
-------------------------------------------------------------


THE   5% THRESHOLD IS AT GRADIENT VALUE    1
AND GRAY LEVEL   22

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS   21 AND   24


 INPUT ELEMENT-VERSION NAME:
                X
NVL2DHISTS  3GN

GRADIENT VALUE      AVERAGE GRAY LEVEL      CUMULATIVE PERCENT
      2                    19                     2.25
      1                    19                    50.62
      0                    19                   100.00


!
!                    .
!                    .
!
-------------------------------------------------------------


THE   5% THRESHOLD IS AT GRADIENT VALUE    1
AND GRAY LEVEL   19

THE MODES OF GRADIENT VALUE 0 ARE AT GRAY
LEVELS   16 AND   21
```

Figure 3.5.2 (continued)

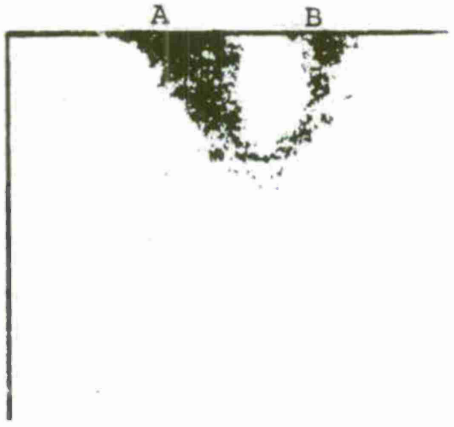


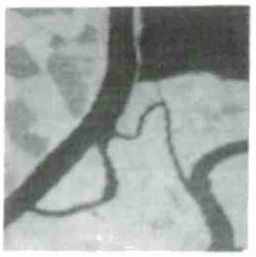
Figure 3.5.3. Ideal 2-D histogram of a scene containing an object and background with noise.

the amount of noise. Several other partitioning schemes which were considered are discussed in [5].

3.5.2 Slice range selection

The methods discussed above predict a single threshold to be applied to an image. For images known to contain a single object class or for small windows, the use of a single threshold is appropriate. However, in general, no single threshold will separate all objects of interest from the background. It is therefore necessary to extend our threshold selection concept to allow the choice of multiple thresholds.

Our approach is to produce clusters of points corresponding to region borders and to associate the average gray level of each cluster with a threshold for the corresponding region. Edge detectors select at each point the maximum difference of averages of adjacent neighborhoods over several directions. By suppressing non-maximum responses normal to the selected direction (i.e., across the edge), thin contours result which appear to surround object regions (see Section 3.7.2). A by-product of this process are points with very low edge value, including values which truncate to zero. Such points correspond to the interiors of homogeneous regions. Figure 3.5.4 illustrates thinned detector responses with region interior maxima included. After thinning, each remaining point is plotted using edge value and average gray level in a two-dimensional histogram. Figure 3.5.5 shows examples of images together with their 2-D histograms based on thinned edges.



a.

b.

Figure 3.5.4a. LANDSAT window of Monterey, CA.
b. Thinned edge detector response (thresholded).

a.

b.

c.

d.

e.

f.

g.

h.

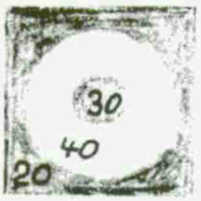
Figure 3.5.5a. Disk (gray level 30) within ring (gray level 40) within background (gray level 20).
b. 2-D histogram of (a) with gray level as x-axis (increasing left to right) and edge value as y-axis (stretched -- increasing from top to bottom). Interior of background is leftmost, topmost cluster.
c. Window containing house.
d. 2-D histogram of (c).
e. Window containing tank.
f. 2-D (stretched) histogram of (e).
g. LANDSAT window of Monterey.
h. 2-D histogram (thinned edge vs. average gray level).

Two types of clusters are produced: interior clusters represent the interiors of regions, edge clusters represent boundaries between regions. The size of a cluster (i.e., the number of points in it) is closely related to properties of the region it describes. Thus interior clusters relate both to the area of the region and to the size of the neighborhood over which the local operations (edge detection, non-maximum suppression) are defined. For small object regions, there may be no points sufficiently far from the object boundary to resist suppression. Thus, interior clusters may be indistinguishable from noise, or may be nonexistent.

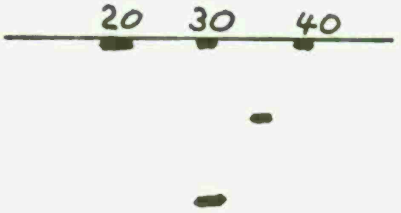
Clusters of points at higher edge values are more likely to be significant (based on our homogeneity assumptions). The size of an edge cluster is therefore related to the perimeter of the surrounded region in the image. Since perimeter increases (roughly, for digital images) as the square root of area, the edge clusters for objects of moderately different areas should, nonetheless, be of comparable size. A priori estimates of size are of use in discriminating true edge clusters from random noise.

Each edge cluster corresponds (ideally) to these interior clusters whose locations can be determined from the location of the edge cluster. Thus a threshold derived from the edge cluster will separate the interior clusters. However, care must be taken not to split an interior cluster at a threshold since this introduces random noise regions.

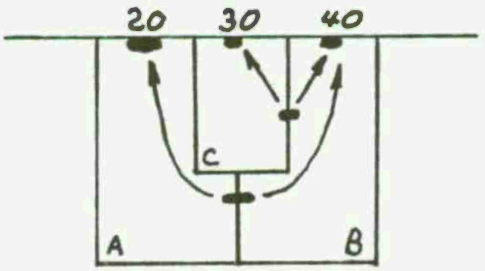
Figure 3.5.6 illustrates a compound decision surface in the 2-D histogram of a multi-object image. For further discussion see [16].



a.



b.

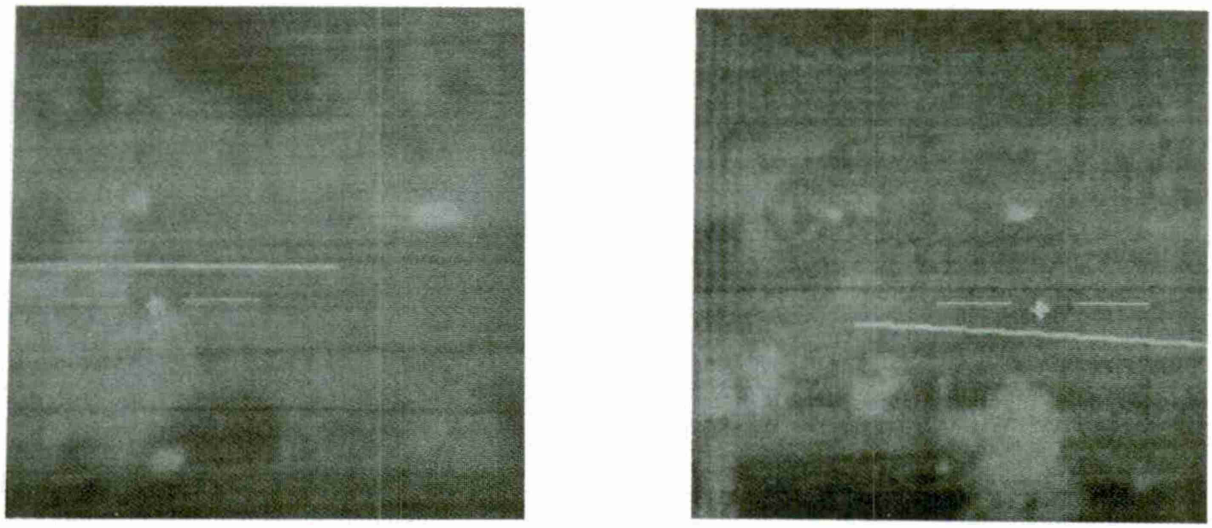


c.

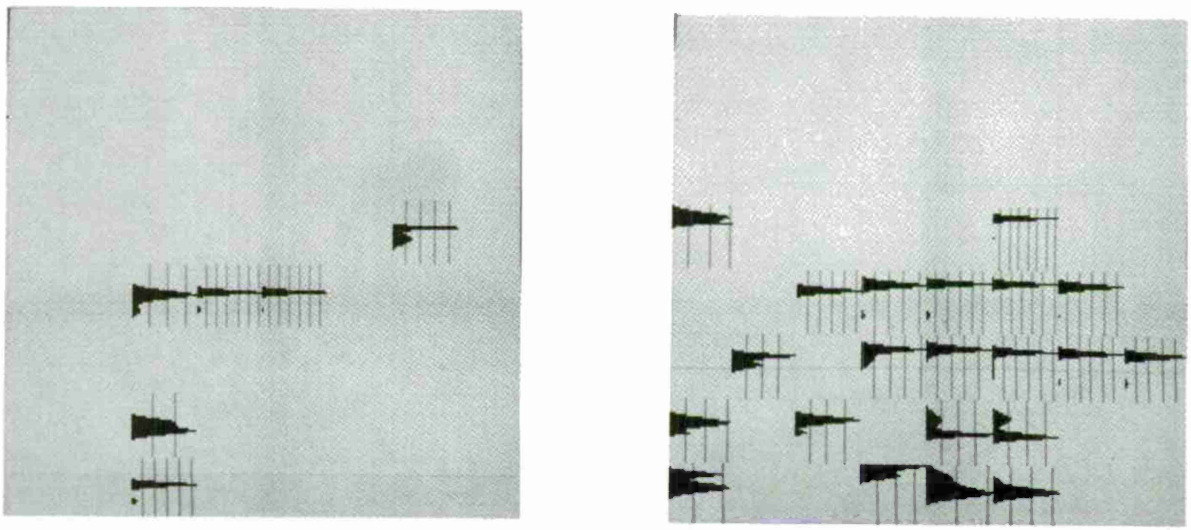
Figure 3.5.6a. Adjacent object regions on background (same as Figure 3.5.5a).
b. 2-D histogram.
c. 2-D histogram partitioned into classification regions.

3.5.3 Variable thresholding

Previous approaches to thresholding apply the same threshold to all points of the image. In [17], Nakagawa adapts the work of Chow and Kaneko [18] to interpolate a best threshold for each point of the image. Briefly, the image is divided into small windows (say 32x32) and a test of gray level histogram bimodality is made for each window. A best threshold is chosen for each bimodal window and thresholds are interpolated to all image points. A binary image results when each threshold is applied to its corresponding pixel value. Figure 3.5.7 compares fixed thresholding and variable thresholding for several FLIR frames. Nakagawa extended this method to allow multiple thresholds (for multi-object adjacencies). Figure 3.5.8 illustrates the results.



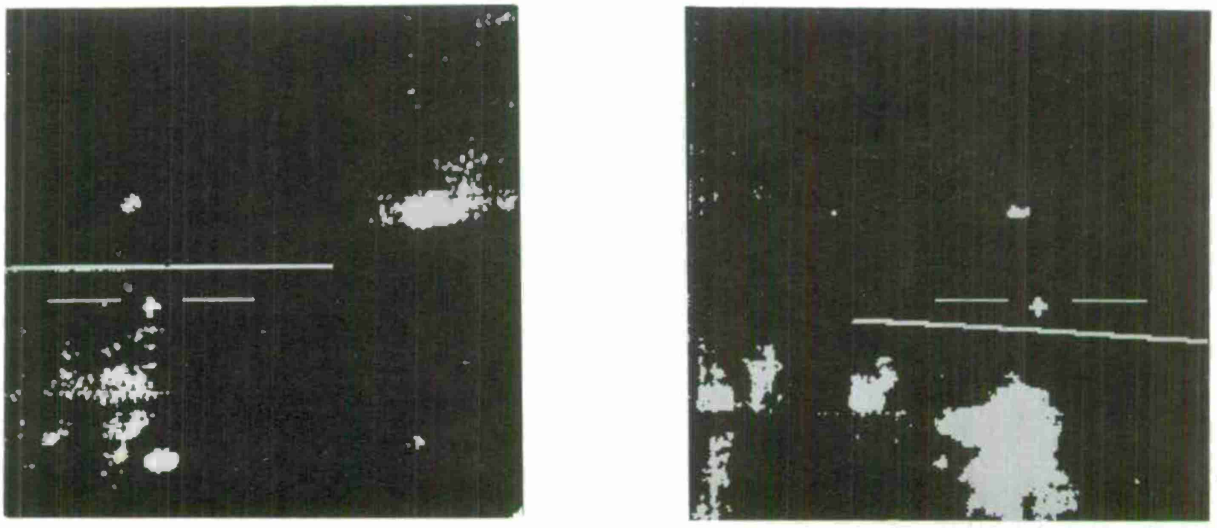
a.



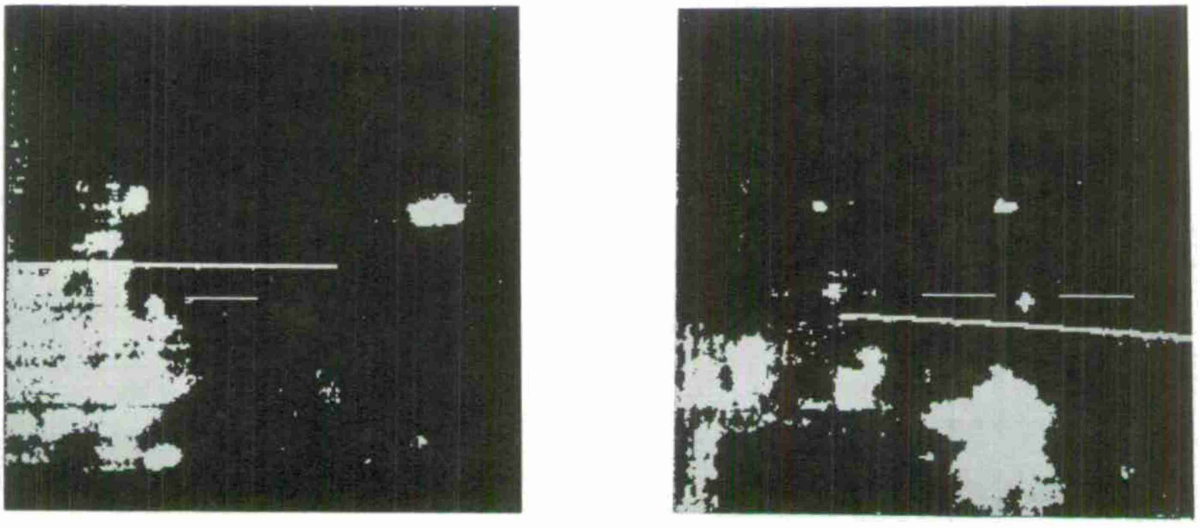
b.

Figure 3.5.7. Comparison of fixed and variable thresholding.

a. Two FLIR frames.
b. Two-Gaussian approximations to those 32x32 window histograms that were judged to be bimodal.



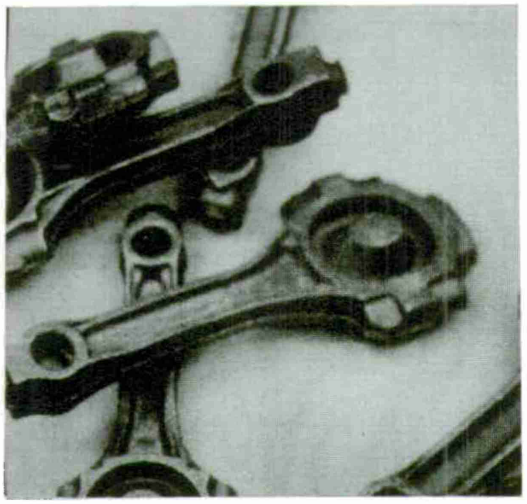
c.



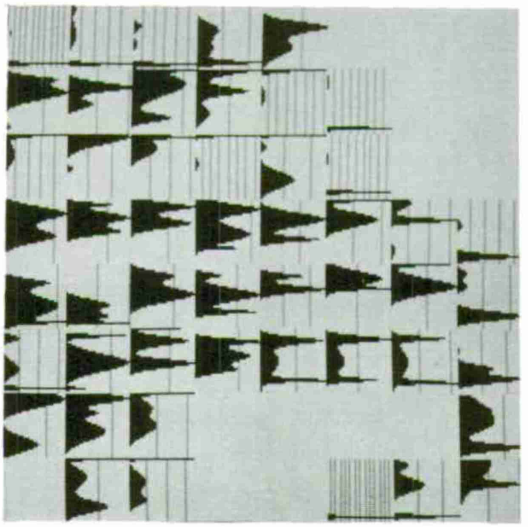
d.

Figure 3.5.7 (continued)

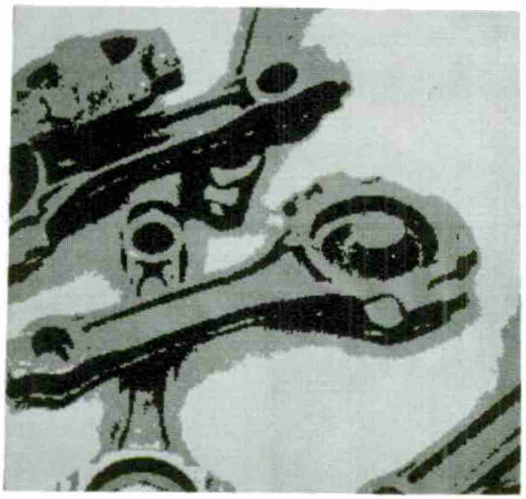
c. Results of applying the interpolated point thresholds to (a).
d. Results of applying a fixed threshold to (a).



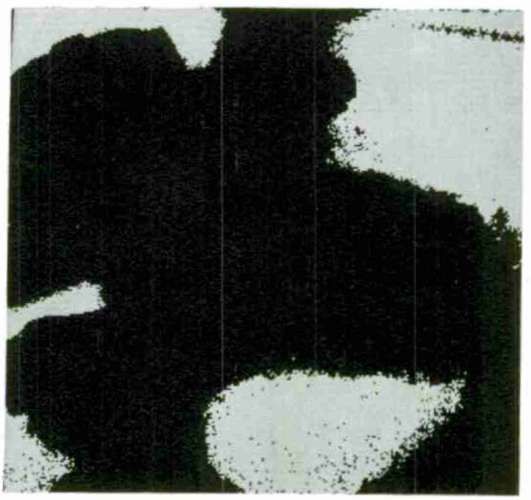
a.



b.



c.



d.

Figure 3.5.8. Comparison of fixed and variable thresholding.

a. Machine parts image.
b. Two- and three-Gaussian approximations to those window histograms that were judged to be bi- and tri-modal.
c. Three level pictures obtained after interpolating the multiple thresholds determined from (b) and applied to (a).
d. Results of applying a fixed threshold to (a).

3.6 Noise cleaning and component labelling

3.6.1 Shrink/expand and min/max

The result of thresholding is a binary valued image. It often contains isolated points and small noise regions which are artifacts of the thresholding and may not be readily visible in the original image. Smoothed images tend to have fewer (but larger) noise regions. One may delete noise regions by postprocessing the thresholded image.

The method consists of multiple applications of two processes: "shrink" and "expand." The purpose of the sequence of shrinks is to shrink objects in a uniform manner so that small or insubstantial objects disappear entirely. The sequence of expands is meant to regrow the remaining shrunken objects to their original size. The result of the shrinks/expands is the elimination of tiny regions (presumed to be noise regions).

Each shrink or expand requires the simultaneous or "parallel" application of a local replacement rule at every point of the thresholded image. The form of the shrink rule is as follows: Eliminate all 1's adjacent to 0's. Zero values are unchanged. Such a rule decreases the number of 1's in the thresholded image; thus, the image "shrinks." Only 1's surrounded by 1's will survive a shrink. The number of successive shrinks determines the minimum diameter of a surviving region.

The expand rule is similar to the shrink rule: rewrite a 0 as a 1 if any of its neighbors are 1's, but

leave 1's unchanged. Thus points adjacent to 1's become 1's, thereby increasing the number of 1's. If we wish to restore objects (that were not eliminated) to about their original sizes, t shrinks should be followed by t expands. Such a shrink/expand sequence produces an image whose 1's correspond to (a subset of the) 1's in the untransformed binary image. Thus, for example, isolated 1's are eliminated, and objects joined by narrow necks of 1's may become disconnected. Also, thin protrusions from a region of 1's will disappear. Figure 3.6.1 illustrates the shrink/expand algorithm for both the 4 and 8 neighbor cases and t = 1,2,3 (the numbers of shrinks and expands used).

A generalization of the shrink rule was formulated to fill pinholes and conserve small region shape as follows: delete a 1 if at least k of its neighbors are 0's (0's remain unchanged). The original shrink rule corresponds to k = 1. If k > 1, it takes more zero evidence to convert a 1 to 0. The generalized expand is analogously defined: Rewrite a 0 as 1 if it has at least k 1's as neighbors (1's remain unchanged).

However, the generalized expand rule is not quite as generous in providing new 1 values, although it does fill pinholes in sufficiently large regions. Figure 3.6.2 provides a comparison for t = 1, 2 and k = 1, 2, 3. The shrink/expand rule with t = 2 and k = 3 applied to each image point and its 8-neighbors provides efficient noise cleaning with

Figure 3.6.1. Effects of iterating SHRINK/EXPANDS (S/E's).

a. Original images - each column is a single image thresholded at four different values.

b. 4-neighbor rule - one S/E

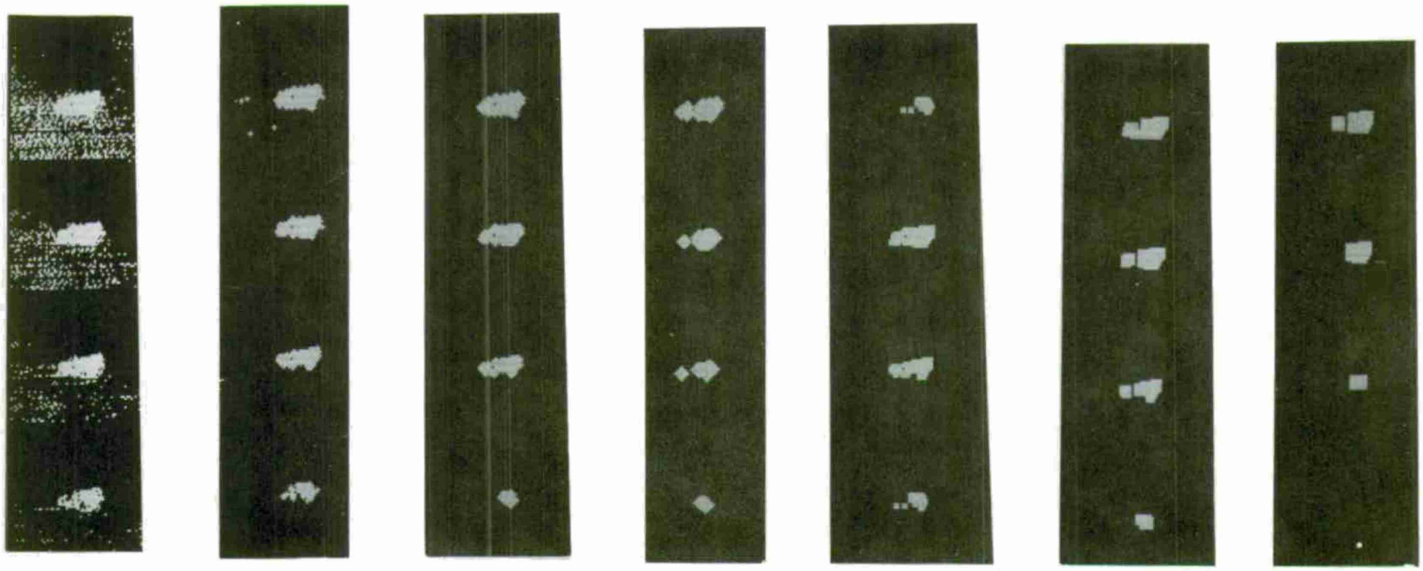
c. 4-neighbor rule - two S/E's

d. 4-neighbor rule - three S/E's

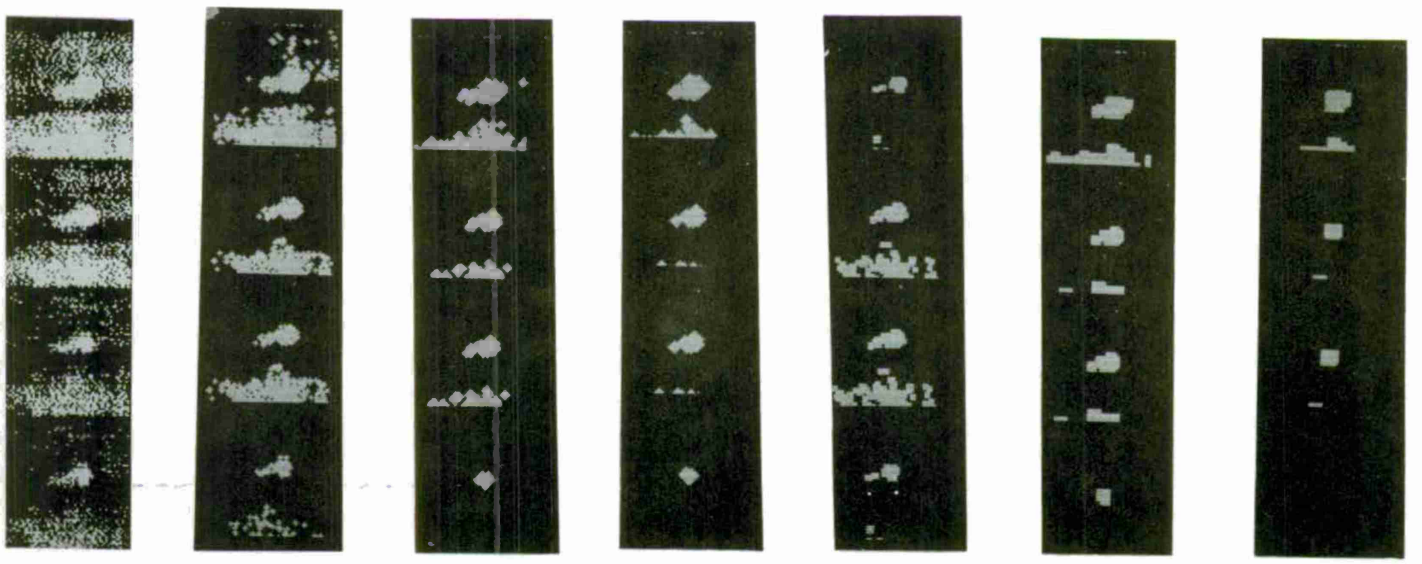
e. 8-neighbor rule - one S/E

f. 8-neighbor rule - two S/E's

g. 8-neighbor rule - three S/E's



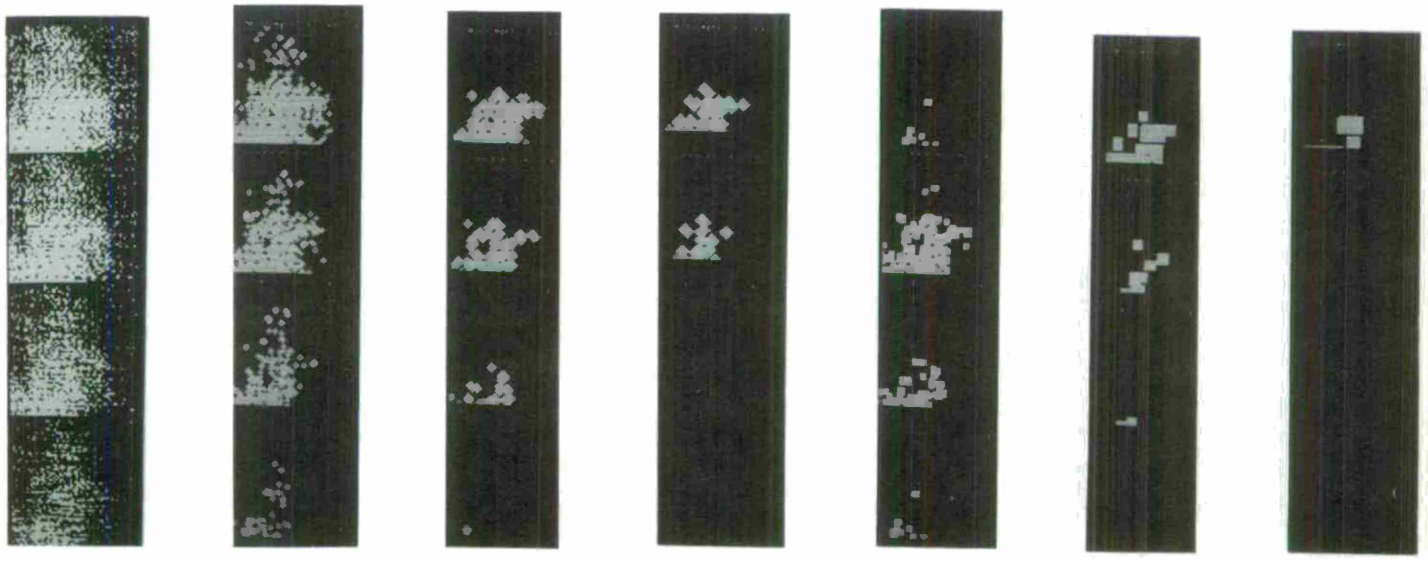
6T a. b. c. d. e. f. g.



6R a. b. c. d. e. f. g.

Figure 3.6.1 (continued)

24A a. b. c. d. e. f. g.



8N a. b. c. d. e. f. g.

Figure 3.6.1 (continued)

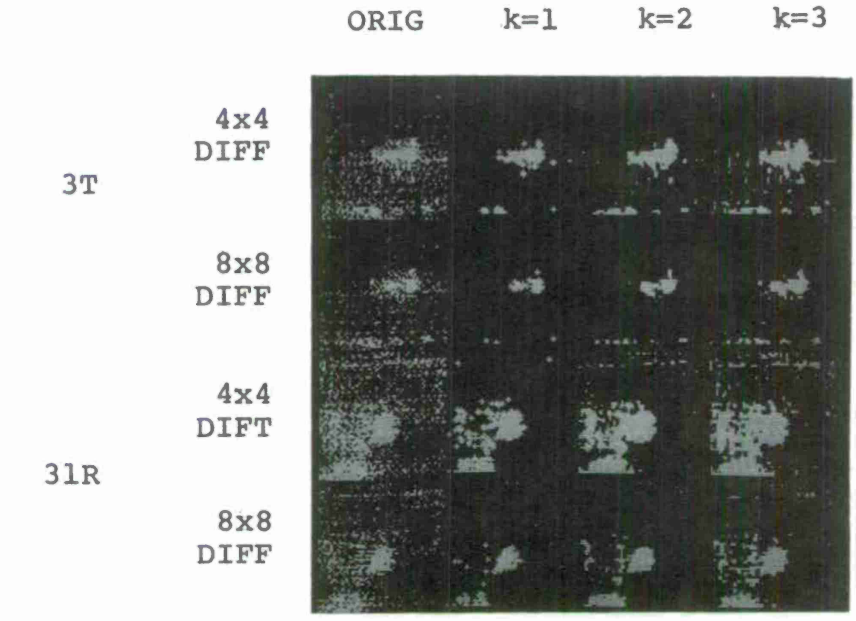
Figure 3.6.2. Leniency in SHRINK/EXPAND definitions for windows thresholded by two methods.

a. 4-neighbor rule, one S/E, k=1,2,3

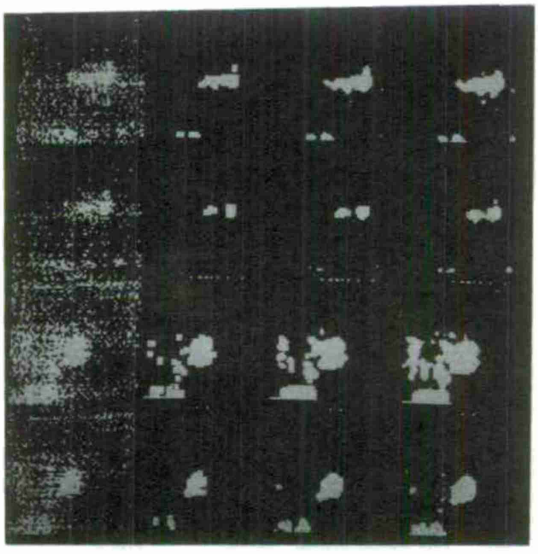
b. 8-neighbor rule, one S/E, k=1,2,3

c. 4-neighbor rule, two S/E's, k=1,2,3

d. 8-neighbor rule, two S/E's, k=1,2,3

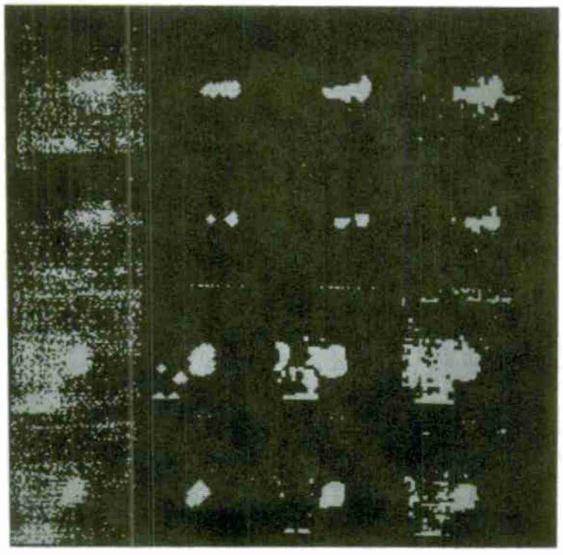


a.

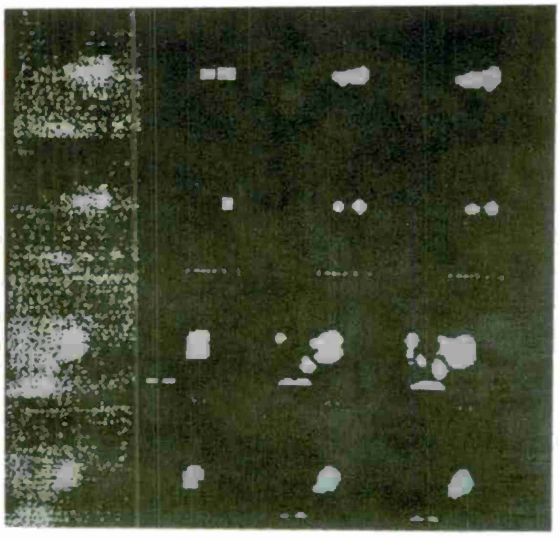


b.

Figure 3.6.2 (continued)



c.



d.

Figure 3.6.2 (continued)

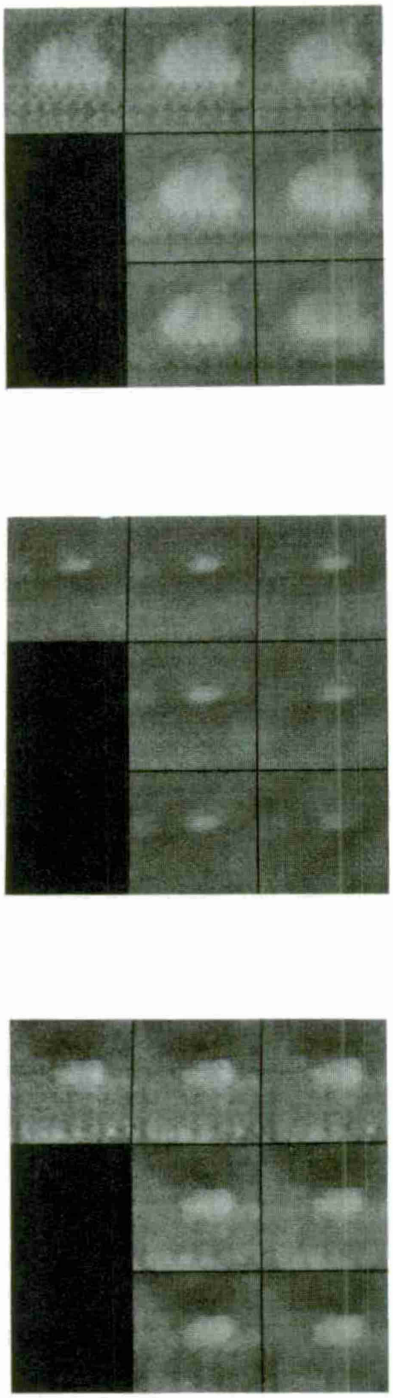
most noise regions eliminated, pinholes filled, and only a modest amount of target shape distortion.

One may further generalize this process for application prior to thresholding. This technique, called "precleaning", involves a sequence of local MIN/MAX operations applied to the gray level image (analogous to shrink/expand applied to a binary image). The resulting precleaned image may now be thresholded as desired. The above threshold regions are as they would have appeared after shrink/expand processing. Figure 3.6.3 illustrates the process. This work is described in [19].

(a)

(b)

(c)



Key:

	4-nbr.	8-nbr.
Original	MIN·MAX	MIN·MAX
	$MIN^2 \cdot MAX^2$	$MIN^2 \cdot MAX^2$
	$MIN^3 \cdot MAX^3$	$MIN^3 \cdot MAX^3$

Figure 3.6.3. Results of applying repeated local MIN and repeated local MAX to three FLIR images. In each part, the upper-left picture is the original; the second column uses 4-neighbor local MINs followed by 4-neighbor local MAXes (1, 2, and 3 repetitions, in the first, second, and third rows); and the third column is analogous, using 8-neighbor operations (i.e., including the diagonal neighbors).

3.6.2 Connected component extraction

The result of thresholding is a binary image. After noise cleaning operations filter this image (as needed), it still remains to aggregate points into identified (labelled) regions. A process which labels the individual disjoint regions in the binary image, in a single raster scan, is well known in the literature [20]. It is described briefly here.

A set of 1's in a binary image is connected if any two points in it can be joined by a path (sequence) of pairwise adjacent points lying in the set. A maximal connected set is called a connected component. The algorithm to be described produces the (unique) decomposition into connected components, labels the individual components, and constructs for each connected component a descriptive feature vector. Although we do not specify the features, it is assumed that they are all extractable from a raster scan using a 3x3 processing window. Additional storage is available to hold the feature values for the components. Section 3.8.2 describes the features.

When a new region is encountered during a raster scan, it is assigned a vector of registers to store its feature values. As the region is being tracked on the same row or continued on the next row, values continue to be accumulated into its feature vector. In order to specify the correspondence between a region and its register vector, a label

is created and assigned to each point of the region which has already been visited. The label will identify the appropriate register vector, usually by some indexing scheme. Region points found to be adjacent to already labelled region points inherit that label and contribute their feature values to its register vector.

Often a region encountered for what is thought to be the first time may on a later row prove to be connected to a previously encountered region. Such regions are called subcomponents. Inasmuch as feature values were being maintained separately for each subcomponent, it becomes necessary to combine the feature values (eventually) and to create a flag that signifies that the two subcomponents belong to the same component. These flags reside in the label equivalence table. This table can be stored either as a bit matrix or as a list.

Since region labels propagate from point to point, we must also keep the labels of those points in the preceding row that are neighbors of unexamined points in the current row, with the labels of those examined points in the current row. The amount of storage necessary for labels of points is thus only a single row.

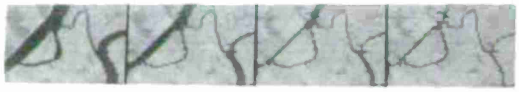
The label assigned to a component should designate whether the component is above or below threshold. If the background is not partitioned into regions (i.e., is ignored) by the algorithm then the data structure becomes simply a

list of above-threshold regions. This is suitable for many applications, e.g., infrared target cueing. In general, though, the containment relation defines a tree structure. It is evident that if two components of a binary image are adjacent then one encloses the other. However, if more than one object-background transition has been detected, one cannot know which encloses which from strictly local information at the time of initial label assignment. The determining condition is "which region terminates first?" The region terminating first is enclosed by the adjacent region. Thus whenever a region terminates, the data structure is updated to reflect the containment relation. When a region is initiated it is entered onto a "active" list -- the list of unterminated regions. At the end of each row, the active list is compared with the list of component labels of the current label row. Any active component whose label does not appear in the current label row is known to have terminated. Additionally, when overlapping regions are combined, the discarded label is deleted from the active list.

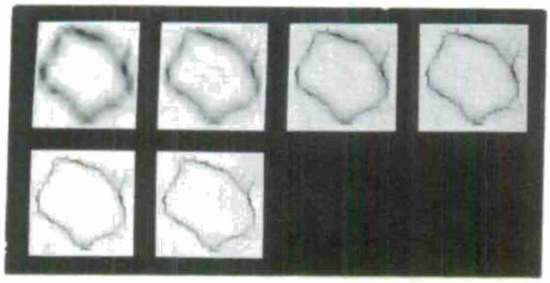
It is possible to modify the above to create a description of each connected component's boundaries. Such a description is called a "chain encoding" and is discussed in [20].

3.6.3 Fuzzy thinning

Objects which are everywhere elongated are often thinned down to a "medial line" for the purpose of extracting thickness-invariant topological features of the objects. The basic strategy for thinning is to iteratively delete border points (but not end points) of an object which do not locally disconnect it. For binary images, various parallel algorithms exist. The recent extension of the topological concept of connectedness to fuzzy subsets allows us to generalize thinning to gray level images [21]. Given thin dark objects on a light background, we define gray level thinning to be the successive replacement of points by the minimum gray level of their neighbors if those changes do not affect the local fuzzy connectedness for any pair of neighbors. The result of applying such an algorithm is a set of high gray level "curves" lying on the ridges and peaks of high gray level in the original picture. If the original picture is noisy there will be many local peaks; so while thinning is defined for unsegmented pictures, a local threshold is necessary to overlook these small noise peaks. Unlike binary thinning, however, we no longer need to distinguish between border and interior points since thinning a homogeneous region will not significantly change the gray level of any point; only a slight smoothing results. The results of experiments with this technique are described in a technical report [22] . See Figure 3.6.4 for examples of this process.



a.



b.

Figure 3.6.4a. Iterations 0-3 of fuzzy thinning on LANDSAT window of Monterey.
b. Iterations 0-5 of fuzzy thinning on the output of an edge detector.

3.7 Superslice

The object extraction task is somewhat simpler for FLIR imagery than for visible-light imagery since the objects of interest (military vehicles) are generally compact regions of (more or less) uniform thermal intensity. For this reason, thresholding has been chosen as an appropriate method of segmenting the scene. However, one can criticize threshold selection schemes on a number of grounds. First of all, if a window contains no object then thresholding it is dangerous, since above-threshold noise regions may often produce probable looking "objects." Secondly, if more than one object is present in the window then a single threshold will not suffice. Thirdly, if an object overlaps several windows then there may be no consistent representation of an object (i.e., no representation using a single threshold). Attempts to divide the scene up into overlapping windows, so that objects of maximal size are guaranteed to lie completely within a single window, answer this last objection at the cost of greatly increased overhead. In any case, the size of the smallest thresholdable region -- as well as the particular threshold chosen -- depends on the window size, the coarseness of the grid, and the type of statistical test used to determine if a region is thresholdable. One would prefer, however, to be able to extract a small region regardless of the clutter and noise beyond its borders.

Another objection to pure thresholding is the presence of noise regions in addition to object regions. Noise regions may be difficult to distinguish when based on size, shape or gray level features. The broader and higher the valleys of the gray level histogram, the more likely that the noise regions will be extensive and numerous.

A final objection concerns the design of optimal thresholding techniques in which the optimality is based on a statistical model of the gray level population. In situations where an object contrasts strongly with the background, there may be a number of thresholds at which the object appears well defined. As the threshold decreases through this acceptable range, each object exemplar is contained within a slightly larger one. Thus although the exemplars may each look reasonable, the optimality criterion for the thresholding does not necessarily choose a "best" exemplar. This is because the optimality condition was based on the whole window rather than on the component corresponding to the object.

For these reasons, a segmentation method which does not require a commitment to a single threshold in arbitrarily chosen regions of an image is preferable. Our method uses thresholding as a means of discovering candidate object regions. Candidates are then accepted or rejected based on the coincidence of an edge map with the region boundary. The surviving object regions are compared with the survivors of other thresholds, and those that best match the

edge map are used to describe the actual objects in the image. Thus, while a number of thresholds are used, only the one defining the greatest coincidence of thresholded region border and (thinned) edge is deemed valid for a particular region. This method can be considered as defining a best exemplar for each object region.

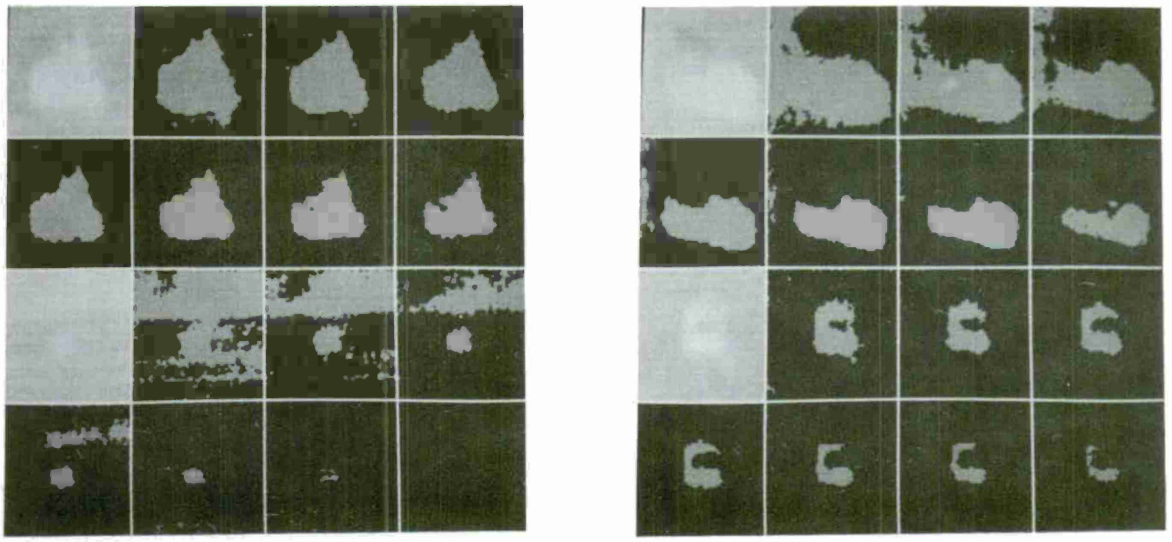
3.7.1 Algorithm

The algorithm consists of several steps as follows: median filtering; extraction of an edge mask by edge detection and thinning; thresholding; forming connected components; and object validity checking. For a given picture, smoothing and edge map extraction need to done only once; whereas thresholding and the subsequent steps are to be performed over a range of threshold sufficient to extract any objects in the picture.

Figure 3.7.1 illustrates the basic concepts involved. Figure 3.7.1a shows several object windows along with a number of possible thresholds for each. Note that it is not at all obvious which threshold is best. However, when the edge map (Figure 3.7.1b) is overlaid on the thresholded picture (Figure 3.7.1c), we have much better guidance. Figure 3.7.1d shows the object region extracted from each window using the method to be described.

A number of steps of the Superslice algorithm have been discussed in previous sections: smoothing (Section 3.3.3), edge detection and thinning (Section 3.4.2), threshold selection (Section 3.5.1) and connected component extraction (Section 3.6.2). However, several problems associated with threshold selection deserve mention:

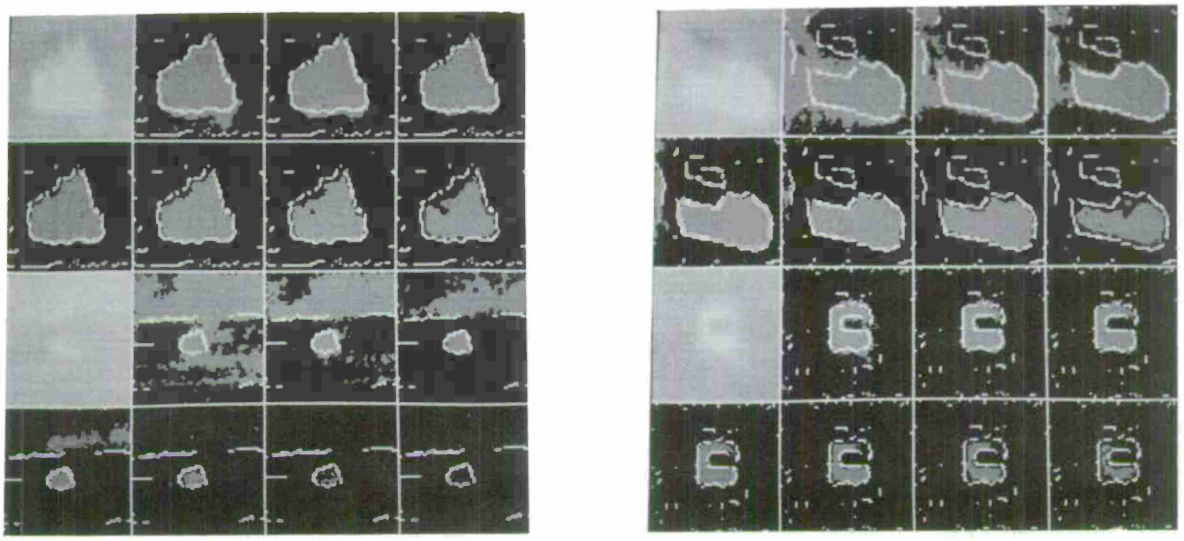
a) The omission of a threshold from consideration increases the probability of missing extractable regions.



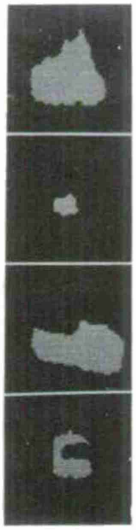
a.

b.

Figure 3.7.1a. Four target windows (large tank, small tank, truck, APC) thresholded at seven different gray levels.
b. Edge maps (thresholded for visibility).



c.



d.

Figure 3.7.1c. Edge maps from (b) overlaid on (a).
d. Object regions extracted by the Superslice algorithm.

b) The greater the number of thresholds considered, the greater the false alarm rate.

c) The speed of the algorithm is approximately linear in the number of thresholds used.

The probability of missing a object region due to the omission of a single threshold is the product of the probability that the scene contains an object region and the probability that the object region is discernible (by the algorithm) at exactly the omitted threshold. Although knowledge of the a priori probability is dependent on a model for the scene (which does not at present exist), experiments have demonstrated that an object region which is discernible at all by the algorithm can be extracted over a range of thresholds -- dependent, of course, on the steepness and homogeneity of the edge region bordering the object. Noise regions, on the other hand, do not tend to persist over a range of gray level thresholds. This tradeoff may therefore be posed as follows: By sampling at every kth gray level, we reduce the workload to a fraction (1/k) without appreciably increasing the false dismissal rate; however, we lose some redundancy in the extracted data which would help us discriminate object regions from false alarms.

The false alarm rate is a function of input window size,

as well as a function of the number of thresholds and the positions of the thresholds in the overall gray level histogram. Certain thresholds are worse than others in producing false alarms -- specifically, those at or adjacent to peaks in the histogram.

After thresholding and connected component extraction, each component must be validated as to whether the extracted region really corresponds to an object in the scene. If one considers validity checking to be a classification process, then one can compute a large number of potential features and, using standard techniques, determine a discriminant function. We have established three heuristics to be of value. One is that objects should be "well-defined," i.e., have discernible borders. Note that not all real-world regions satisfy this constraint. For example, in LANDSAT scenes, forests, urban areas and clouds can blend into their surrounds with no discernible edge. The second heuristic is that an object's interior should "contrast" with its surround. In this study, contrast is based on gray level difference. However, other local features including texture measures are worth considering as defining object interior. The third is that the region size lie within an acceptable range. The size test is applied first, eliminating any region with fewer than 20 or more than 1,000 points.

"Well definedness" of a region is measured by the percentage of border points which correspond spatially to

(match) actual edge points in the edge map. "Contrast" is measured by the absolute difference of average gray level between the border region of the component and its interior. Figure 3.7.2 shows a scatter plot of these two features for the regions extracted from a set of windows. A reasonable discriminant based on these two features appears to be: match > .5 and contrast > .6 -- i.e., at least 50% of the border matches the edge map, and the contrast is at least .6 gray levels (out of 64). Note that neither feature is by itself reliable enough to discriminate noise regions from object regions. Optimal discriminants may be computed based on several models. Regardless of the particular model chosen, the discriminant value can be interpreted as a "score" for the component. Components with very low scores are discarded as pure noise. In practice, we have used the match measure as a score for objects which were above the pure noise threshold.

The score is important in comparing (nested) object regions corresponding to the same object. When an object is thresholdable at gray levels $t_1 > t_2 > \ldots > t_k$, this gives rise to k connected components, $C_{t_1} \subseteq C_{t_2} \subseteq \ldots \subseteq C_{t_k}$. Since each C_{t_i} represents the same object, we call each an "exemplar." In general, we wish to select a single exemplar as the best representative of an object. The score provides a criterion for selecting among exemplars. Thus, one could choose the

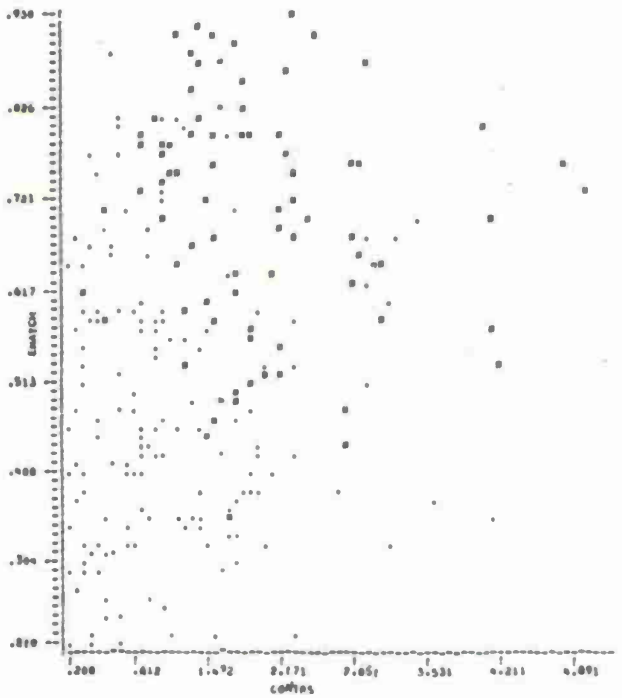
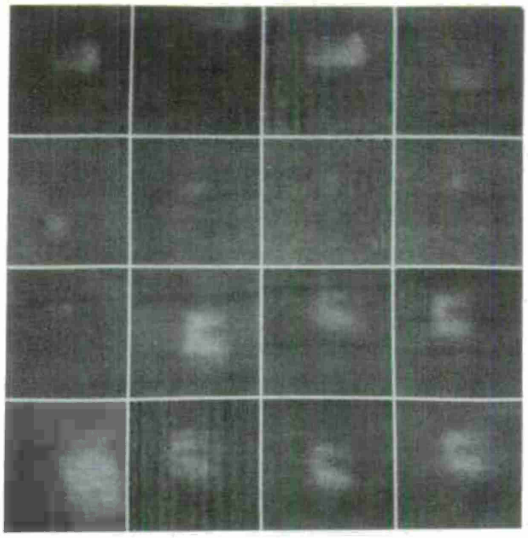


Figure 3.7.2. Scatter diagram plotting well-definedness against contrast for a set of noise regions (plotted as periods) and object regions (plotted as hash marks).

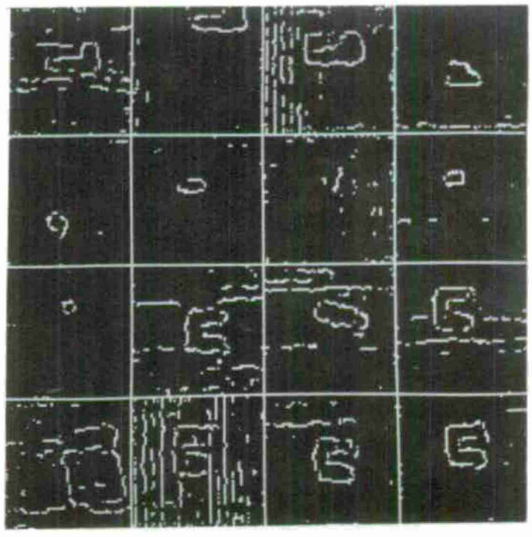
exemplar C_{t_j} with the highest score. It is not always easy, however, to determine the nested sequence $\{C_{t_i}\}$. In particular, if one object thresholdable at gray level t is contained within another thresholdable at gray level t' < t, then regardless of the comparative difference between the two scores, we would want to retain C_t and $C_{t'}$. This situation can be handled by assuming that nested components whose areas are sufficiently different (say, 50% change in size) correspond to different (although nested) objects. In thermal images, this might correspond to a warm vehicle with a hot engine compartment, or to a vehicle on an asphalt road. The results of applying the algorithm to a set of 16 APC windows are illustrated in Figure 3.7.3. Note that in almost all cases (the negative image was not processed), the resulting labelled images contain the target regions (as well as other regions).

In summary, the algorithm for region extraction consists of the following steps:

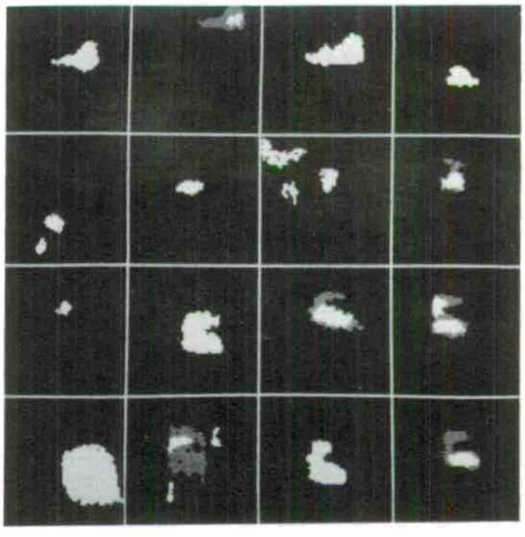
1. Smooth the image, if necessary (to promote clean thresholding).
2. Extract a thinned edge picture.
3. Determine a gray level range for thresholding.
4. For each gray level in the range:
    a. Threshold the smoothed image.
    b. Label all connected regions of above-threshold points.



a



b



c

Figure 3.7.3a. Sixteen APC windows.
b. Edge maps (thresholded for visibility).
c. Object regions extracted by the Superslice algorithm.

c. For each connected region:

i. Compute the percentage of border points which coincide with significant thinned edge points.

ii. Compute the contrast of the region with the background.

iii. Classify the region as object/non-object based on the size, edge match and contrast.

5. Construct the canonical tree for the set of object regions based on containment.

6. Prune the containment tree by eliminating adjacent nodes which are too similar.

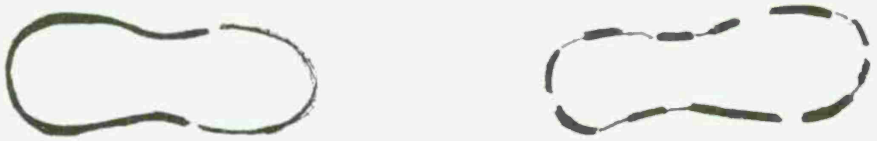
3.7.2 Conformity - a measure of region definedness

The Superslice algorithm relies on the heuristic that thresholded object regions are distinct from background because they contrast with their surround at a well-defined border. The coincidence of high contrast and high edge value at the border of a thresholded region is an example of the use of convergent evidence supporting the assertion of the object region. The definedness of the border may be evaluated as the percentage of the border points which coincided with the location of thinned edge (locally maximum edge response). Thus a match score of 50% means that half the border points are accounted for as being on the edge. However, it does not mean that the matched points adequately represent the object. Figure 3.7.4 illustrates two cases of 50% match. (Matched points are indicated by thick strokes.) Clearly, the second case is a better representation than the first.

The traversal of the border of a thresholded region induces an ordering on the matched points. Let $r_1,\ldots,r_n$ be the runs of matched points encountered during a border traversal. By connecting the proximal ends of runs along the traversal, one creates a polygonal approximation to the thresholded region. We define "conformity" as the measure of match of the polygonal approximation to the thresholded region. High conformity means that the region is well-represented by its approximation regardless of the actual percentage of matched border points. Figure 3.7.4a illustrates

low conformity; while Figure 3.7.4b shows good conformity.

Conformity is evaluated as the ratio of the absolute difference in area (between the two polygonal representations) to the area of the threshold region. Experiments have indicated its utility as a feature for discriminating noise from objects. A quantitative study of its discrimination value is described in Section 3.9.4.2.



a. b.

Figure 3.7.4a. Contour whose matched edge points (thickened strokes) exhibit poor conformity.

b. Contour showing good conformity.

3.7.3 Hyperslice - An algorithm for recursive region extraction

The algorithm (Hyperslice) described here is an amalgam embodying the recursive control structure of Ohlander [3] and the object extraction techniques of Superslice. Hyperslice consists of the following steps [24]:

1. Preprocessing - image smoothing, thinned edge map extraction.
2. Initialize the extracted region mask (ERM) to the empty mask. Initialize the available points mask (APM) to the entire image.
3. Compute histograms for all feature images based on the APM.
4. Determine a "best" slice range over all current histograms and slice the corresponding image.
5. Generate submasks for regions satisfying the Superslice criteria. Add them to the ERM; delete them from the APM.
6. Apply algorithm steps 3-5 recursively to the background set (APM). The algorithm should also be applied recursively to each submask added to the ERM, since the extracted region may be a union of regions discriminable by some other feature.

Several comments are in order. First, the slice ranges chosen for Hyperslice should be rather liberal (i.e., extending beyond valley bottoms in the histogram), since

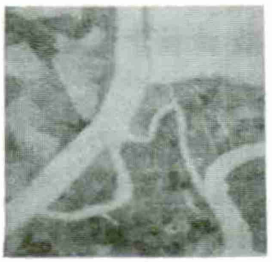
points not corresponding to well-defined regions will be returned to the APM. The resulting histograms appear more natural (not "carved-out") for this reason. Secondly, the resulting decomposition is order-dependent, i.e., different results may be obtained if the order of selection of slice ranges is changed. If two adjacent regions in the image contribute adjacent peaks in the histogram, then points in the intersection of the overlapping slice ranges will generally belong to the shared edge region. Whichever region is sliced first will tend to accrete more of these points. Since these points lie at or near the true edge, they tend to increase the edge match criterion for that region. Once they are removed from the APM, they are not available to the adjacent region. Consequently, the edge match criterion of the adjacent region may suffer. This is most likely to occur for adjacent regions which lack a strong common border. The 2-dimensional histogram approach in [16] can detect adjacency along weak borders. In practice, the edge match criterion is relaxed somewhat from demanding actual coincidence to allowing proximity (e.g., a region border point adjacent to a thinned edge point is counted as a match).

The algorithm has been implemented as an interactive system of programs. Several examples illustrate its ability to segment images based on gray level alone (i.e., no other features were used to aid the segmentation). Figure 3.7.5 depicts a window of an ERTS frame of the Monterey area in

California. The water area contrasts sharply with the land and very little noise is extracted and subsequently returned to the APM. The subsequent slices extract light and dark fields which contrast with the undifferentiated background region.

The second example is derived from Ohlander's house scene. The average of the three color bands provides the gray-scale. The resulting image has been smoothed by 3x3 median filtering. The first slice range extracts the sky regions and the bright crown of a bush. Next the shadow regions appear along with the bushes. The somewhat darker grass is extracted in the third slice range. Finally, the brick is extracted. Figure 3.7.6 illustrates this sequence.

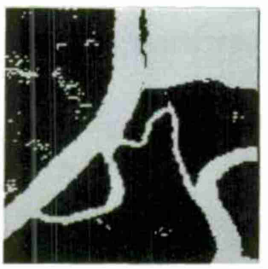
Images such as the Monterey and house images are difficult to analyze since regions need not be well defined due to the complexity of light reflections and shadows. Nonetheless, this algorithm provides a mechanism for retrieving those regions which <u>are</u> well-defined.



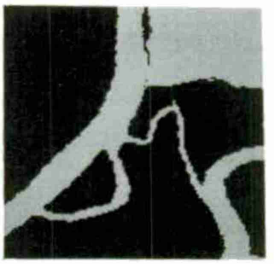
a.

b.

c.



d.

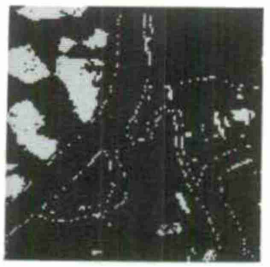


e.

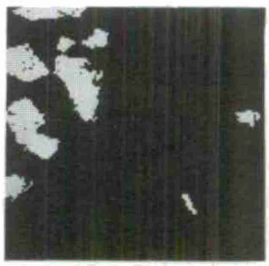
Figure 3.7.5. Recursive region extraction on Monterey image.

a. LANDSAT window.
b. Edge map.
c. Histogram of (a), with selected slice range indicated.
d. Mask of slice range. Within range points are white.
e. Extracted regions mask.

f.

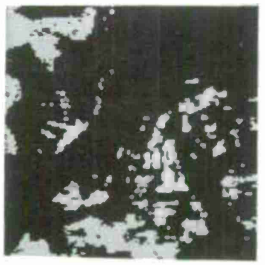


g.

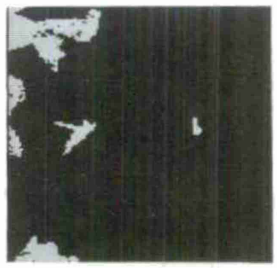


h.

i.



j.



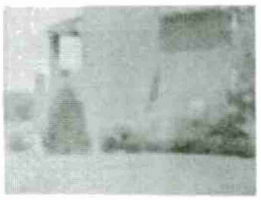
k.

l.

m.

Figure 3.7.5 (continued)

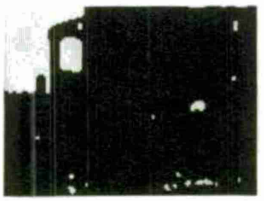
f. Histogram of remaining points after deleting extracted regions of (e).
g. Slice range mask.
h. Extracted regions mask.
i. Histogram of remaining points.
j. Slice range mask.
k. Extracted regions mask.
l. Histogram of remaining points.
m. Mask of remaining points.



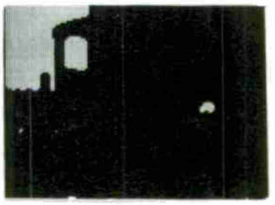
a.

b.

c.

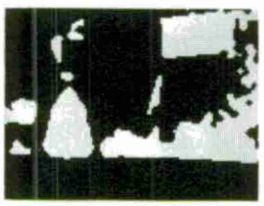


d.

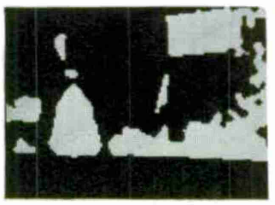


e.

f.



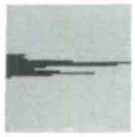
g.



h.

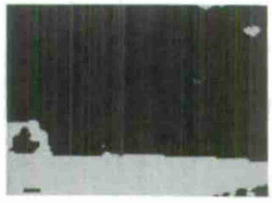
Figure 3.7.6. Recursive region extraction on house image.

a. House window.
b. Edge map.
c,f,i,l,o. Histograms after successive deletion of extracted regions. New slice ranges are indicated.
d,g,j,m. Slice range masks.
e,h,k,n. Extracted region masks.
p. Mask of remaining points.



i.

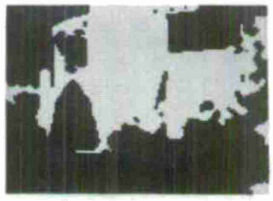
j.



k.

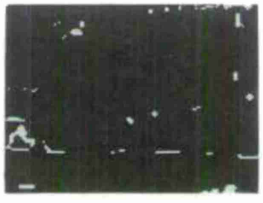
l.

m.



n.

o.



p.

Figure 3.7.6 (continued)

3.8 Feature extraction

3.8.1 Feature design

In this section, as in most work dealing with pattern classification, a "feature" is taken to be some numerical quantity which can be calculated for each object to be classified. ("Shape" is not a feature, since many features, such as height/width, measure characteristics of the shape.) To be consistent with a high processing rate throughout, all features used in this study are based on accumulatable quantities. That is, a number of crude features have been chosen (listed in Table 3.8.1a) which are defined at each pixel. The value of any of these features for a region is just the sum of the values over all the pixels of the region. These crude features can be accumulated as the image is being segmented, and are therefore immediately available for any region as soon as it has been completely extracted. The descriptive features actually used are simple functions of these accumulatable quantities, so that once any region has been extracted, brief calculations produce all the information required for classification of that region, with no further reference to the original image. One additional feature, "conformity," has been obtained for many of the images. This feature requires rather more post-processing after region extraction, and is included as a nearly optimum measure of one region characteristic which should be of importance in target detection: cooccurrence

a. Accumulatable features per connected component

	Symbol	Meaning
1.	N	Area
2-3.	SX,SY	$\Sigma X, \Sigma Y$ - first moments
4-6.	SX^2, SY^2, SXY	$\Sigma X^2, \Sigma Y^2, \Sigma XY$ - second moments
7.	P	Perimeter point count
8.	E	High edge point count
9.	SPE	Total edge value on the perimeter
10.	SIG	Total interior gray value
11.	SPG	Total perimeter gray value
12-13.	SG, SG^2	Total gray level, total squared gray level

b. Intermediate quantities

1.	X_{AVE}	$4 * \sqrt{SX^2}$
2.	Y_{AVE}	$4 * \sqrt{SY^2}$
3.	R^2	$SX^2 + SY^2$
4.	V	$SG^2/N - (SG)^2/N^2$

Table 3.8.1. Features.

c. Recognition features

1.	h/w	Y_{AVE}/X_{AVE}	shape
2.	(h/w)'	$\lvert X_{AVE}-.8*Y_{AVE}\rvert/\sqrt{X_{AVE}*Y_{AVE}}$	
3.	(h*w)/A	$X_{AVE}*Y_{AVE}/N$	
4.	(h+w)/P	$(X_{AVE}+Y_{AVE}-4)/P$	
5.	diff	$(SX^2-SY^2)/R^2$	
6.	skewness	$\lvert SXY\rvert/R^2$	
7.	asymmetry	$((SXY)^2-SX^2SY^2)/R^4$	
8.	SDEV	$\sqrt{V}$	brightness
9.	Gray level difference	SIG/(N-P) - SPG/P	
10.	E & P	(Number of perimeter points at high edge local maxima)/P	
11.	E_p	SPE/P	

d. Special features

1.	conformity	(See Section 3.7.2)

of the region perimeter and points of high brightness gradient. This gives a useful standard for measuring the adequacy of the rapidly calculated feature (E&P, in Table 3.8.1c) which is used as a measure of the same property.

A decision rule is effectively a mapping from the feature space to a lower-dimensional space (the decision space) in which each point is associated with a fixed class. While this structure is very general, commonly used decision rules are very severe specializations of this general scheme. Usually the initial mapping is produced by a set of polynomial functions on the features, one function for each dimension of the decision space. Within this space, the class regions are usually separated by planar boundaries. Thus, the Fisher method utilizes a single linear mapping onto the line, which is bisected by a point (at the Fisher "threshold") to establish the two class domains. Specialization of decision rules places sharp restrictions on what constitutes an appropriate feature.

To discriminate tanks from trucks, a naive observer might point out that one need only examine the shapes. One more familiar with computational measures would recognize that the shape of an object involves a great many features, but might suggest that the height-to-width ratio would be one useful feature. However, height-to-width, width-to-height, log(height-to-width), etc. are all quite distinct features, one of which may be highly effective in the desired

decision while others may be totally useless. Useful features must thus satisfy a number of conditions, some of which are general, the others being imposed when particular simple decision rules are to be applied. The present classification study has considered linear and quadratic classifiers, a decision space with no more dimensions that the number of classes, and simple boundaries for each class within the decision space. Several levels of restriction on the features to be used with such a classifier can be stated:

1. Each feature must exhibit a different distribution for each of at least two classes.
2. The classes should tend to fall in different value ranges for each feature, since class assignments in the decision space will be to connected regions.
3. When the classifier utilizes sample means and variances to estimate parameters for the mapping (as those used here do), the true feature distributions of each class should be unimodal, approximately symmetric about the mode, and with a minority of points contained in the wings of the distribution.
4. For use with <u>linear</u> classifiers, each feature should have a distinctly different mean for at least two classes. For use with <u>quadratic</u> classifiers, it is only necessary that some range of values tend to characterize one class, while

the other class predominates on the complement.

Despite these "rules" for good features, it should be noted that for a multi-feature decision scheme, none of these rules is essential. However, only when some of the features are very strongly correlated can the above principles be violated without destroying the classification, and while this situation is not necessarily to be avoided, it makes interpretation of decision rules much more difficult. Moreover, as a practical matter, features which fail to have the above properties normally turn out to be ineffective (or worse, countereffective) when employed in automatic classification. Since one is not really restricted in the particular form of the features to be used (but only in the underlying characteristic being represented) one may as well assure that the features being considered are, as far as possible, individually effective means of class discrimination.

Finally, one more restriction should be stated.

5. The features should not reflect characteristics which effectively delineate the sample classes, rather than the true classes.

This, of course, is the familiar failing of "small" samples, but may appear even in apparently large enough samples. In our data base (Section 3.9.1), several such "extraneous differentiations" did arise. In cases where a large number of features are employed in a classifier, there must always be doubt about whether condition 5 will hold. It is this

condition, more than any other, which restrains the number of features which can usefully be included in a classifier. If an arbitrarily large number of features are measured for a particular set of classified samples, it is virtually certain that spurious characteristics will allow them to be well separated by a decision function based on those features, but there is no reason to expect anything other than random classification of new samples. The problem is sufficiently pervasive that a simple means of dealing with it could almost be elevated to a principle:

> 5'. Features should be included in a classifier only if they identify true differences between the classes more than they do spurious differences between the samples.

While the above rule may seem obvious, it is important to realize that including additional features that do not discriminate between classes makes the classifier worse, as the features may very well distinguish the class samples, even though they do not distinguish the classes. (Self-classification of the training set improves, while classification of independent test sets degrades.) Class differences must be effectively reflected in the feature to make it safe to use. "Height-to-width" ratio is a dangerous feature to include in a linear classifier for target vs non-target since its mean values for target and non-target classes may not be greatly different (though the distributions may differ greatly),

so that small spurious differences in sample means may produce most of the "strength" of the feature. In a quadratic classifier, however, the problem would be much less severe, since the discrimination provided by the feature more nearly matches the requirements of the decision function employed.

3.8.2 Computation

The principal attributes of image regions which can be used to identify them are shape and relative brightness. Corresponding locally accumulatable properties are pixel coordinates, and functions of them, and gray level, and functions of it. Additional information can be obtained from the contrast between the region and its surround at the region boundary. One can know as one examines each image point whether it is in the interior of a region, on the region boundary, or in the background. Statistics of interest can therefore be accumulated separately for these classes. Finally, the pre-computed edge value (gray-level gradient) is associated with each point, and these values may be accumulated or may be used to index subsets of points (e.g., "high edge" points) for which other quantities may be accumulated separately. The accumulated features actually used are all of one or the other of the above types, and were listed in Table 3.8.1a.

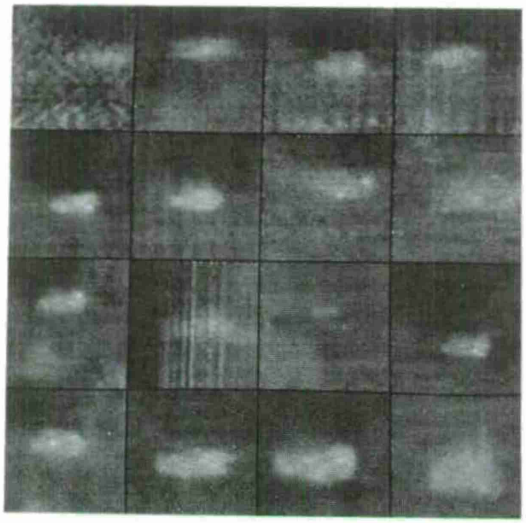
The features calculated for use in classification studies are given as Table 3.8.1c-d. They are further divided into two groups -- those that are purely shape measures, and those that depend in some way on the brightness of the region (or some part of it). Many of the functions appear to be straightforward measures of significant characteristics, but others seem less straightforward. The criteria for choosing the specific functional forms used are discussed in Section

3.9.4. A discussion of the relative utility of the features appears in that same section.

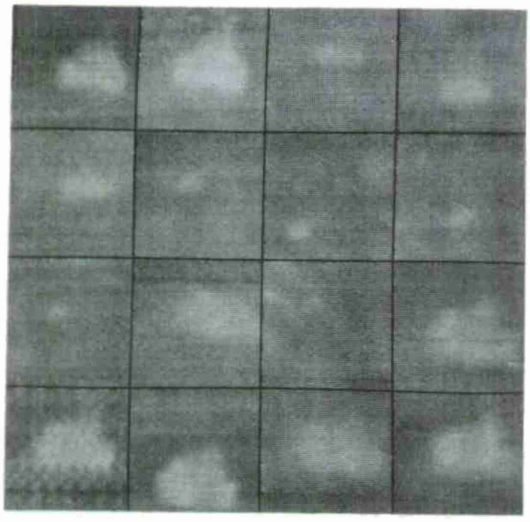
3.9 Region Classification and Experimental Results

3.9.1 Data base description

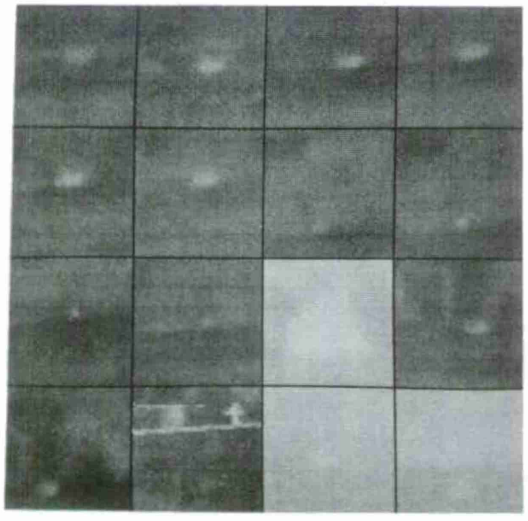
For a description of the complete "NVL" data base and its ground truth see [1]. From it a set of 174 128x128 windows were selected, extracted, requantized, median filtered and sampled 2 to 1. The set consists of 164 target windows (75 tanks, 34 trucks, 55 APC's) and 10 non-target (noise) windows. Figure 3.9.1 displays this set of windows and their identifiers.



1T 2T 3T 4T
6T 8T 9T 10T
11T 12T 13T 14T
15T 16T 17T 21T



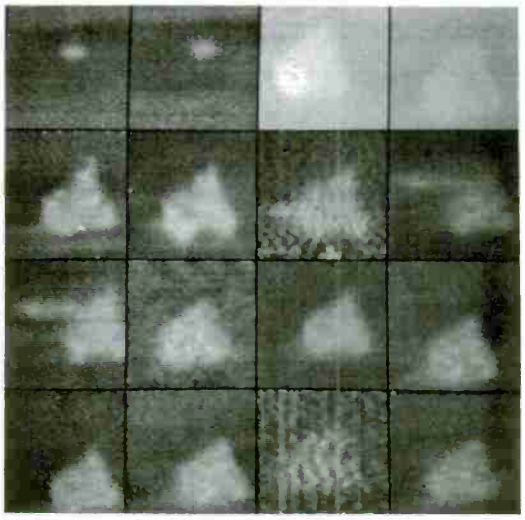
22T 24T 26T 28T
31T 32T 33T 34T
35T 38T 40T 42T
43T 45T 46T 48T



50T 51T 52T 53T
54T 55T 56T 57T
58T 59T 61T 62T
63T 64T 65T 66T

Figure 3.9.1. NVL data base consisting of 164 target windows and 10 non-target windows.

a. 75 tanks.

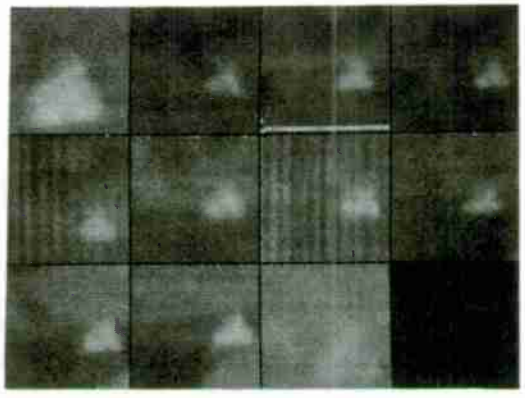


68T 69T 73T 74T

75T 76T 78T 79T

80T 89T 92T 95T

99T 105T 109T 110T

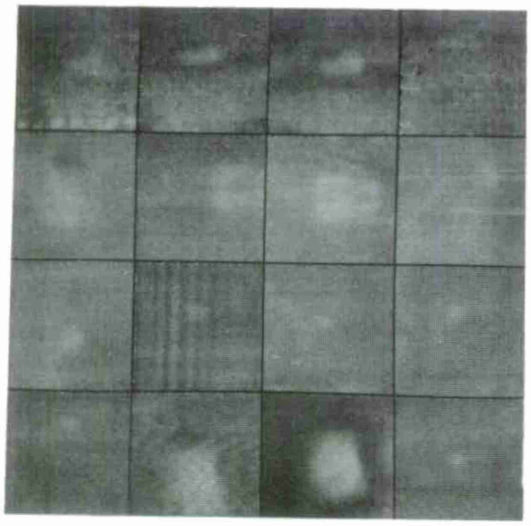


114T 122T 123T 124T

125T 126T 127T 128T

129T 130T 131T

Figure 3.9.1 (continued)

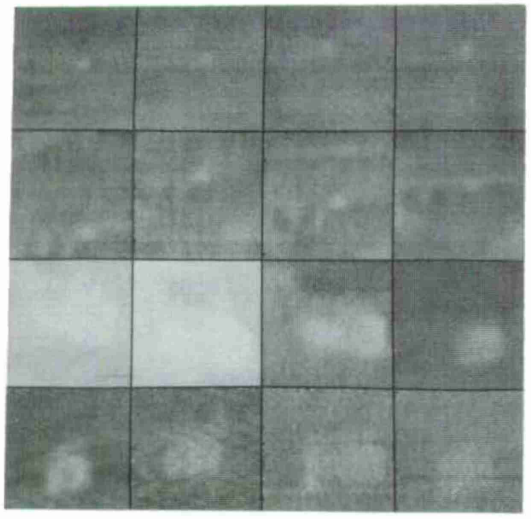


3R 4R 6R 9R

18R 22R 24R 26R

31R 32R 33R 34R

35R 41R 47R 51R

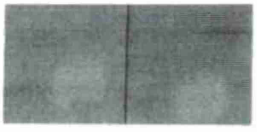


52R 53R 54R 55R

56R 57R 58R 59R

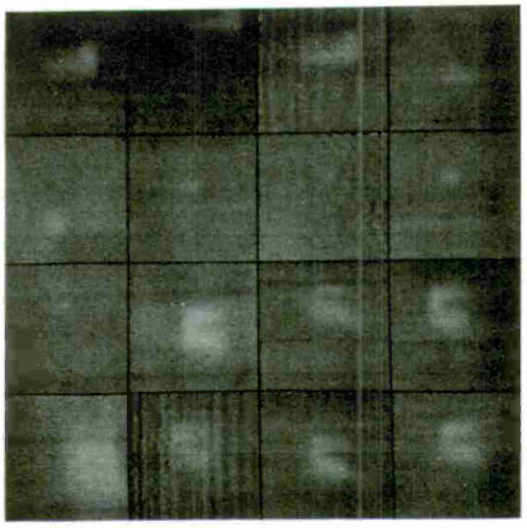
71R 72R 77R 100R

104R 109R 132R 133R

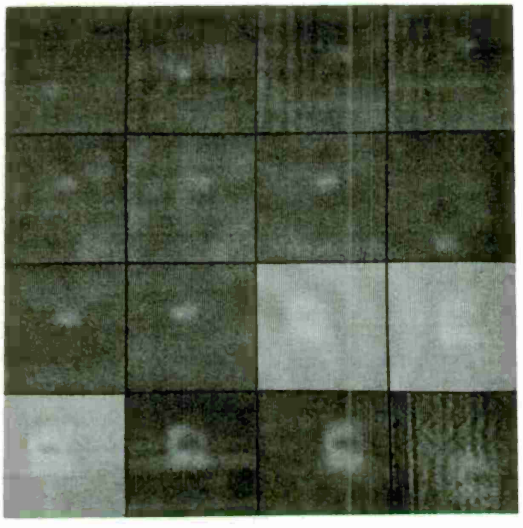


134R 135R

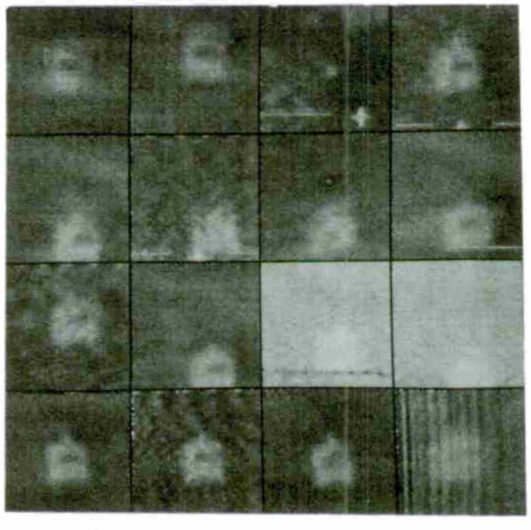
Figure 3.9.1 (continued)
b. 34 trucks.



21A 22A 24A 27A
28A 32A 33A 34A
35A 37A 38A 42A
44A 45A 46A 48A

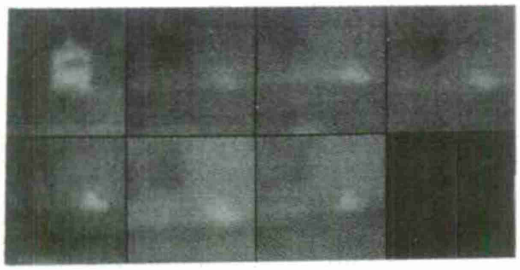


50A 51A 52A 53A
54A 55A 56A 57A
58A 59A 61A 73A
74A 75A 76A 78A



79A 80A 86A 90A
91A 93A 94A 96A
97A 98A 101A 102A
111A 112A 113A 114A

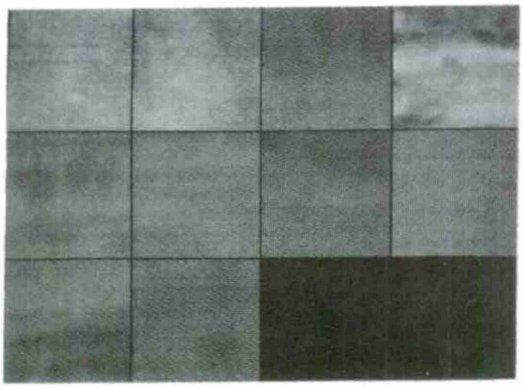
Figure 3.9.1 (continued)
c. 55 APC's.



115A 122A 123A 125A

127A 129A 130A

c. APC's (continued).



2N 8N 14N 20N

26N 32N 38N 44N

50N 56N

d. 10 non-target windows.

Figure 3.9.1 (continued)

3.9.2 Overview of classification

There are two general approaches to classification of objects into a preassigned set of mutually exclusive categories. The first might be called "semantic" classification. Each category is examined for particular characteristics which distinguish its members from those of every other category being considered. These characteristics are used to identify each object submitted for classification. (Difficulties, of course, occur if an object has none of the "key" characteristics, or has "key" characteristics suggesting more than one classification. Such an occurrence indicates that the classes suggested simply do not include everything within the domain of interest, or are not truly mutually exclusive -- at least as defined by the set of "key" features.) This is a form of classification which is ubiquitous in human experience. Unfortunately, in many cases of practical importance, the objects to be classified cannot be characterized by properties which will always be observed within one class, and never in any other class. If the classes really are well-defined, this difficulty may arise because of the need to classify using noisy or poorly resolved data. It may also occur because characteristics quite plain to human observers may defy expression as calculatable quantities (one vehicle may be "sleek and speedy looking", another "squat and out-of-date"). For whatever reason, when such incompletely characterized problems arise,

a method is required which provides a computable "best guess" classification. All such methods accept a number of (usually numerical) features which are assumed to be relevant to the classification intended. The distribution of these features for a large number of objects whose identity is already known is then used to provide a rule which assigns a class to an object given the n-tuple of features measured for that object. Typical rules of this sort are simple polynomials over the features, whose values are used to determine the class assignments.

"Statistical" classification finds the best rule for a fixed class under some (usually very restrictive) assumptions about the way the features ought to be distributed. Since the data available in this study appear not to provide enough resolution to produce a semantic classification, we have utilized a procedure which includes a statistical classification component. A completely statistical classifier was not used, however. The full procedure consists of a semantic pre-classification of regions which could not represent targets, followed by a statistical classification of the "reasonable" regions. This approach was chosen primarily to ensure greater robustness in the resulting classification scheme, as will be discussed more fully below.

Finally, it is important to analyze the types of errors made by a classifier. For example, a well-behaved classifier should be wrong more often on distorted images

than on undistorted ones. This type of performance may be tested by training a classifier of the same type on a "training set" of half the samples, distributed evenly through the classes. The resultant classifier can then be used to reclassify the whole data set. If the "training" and "test" results are similar, then the classifier is judged fairly stable. If the results are good, then the classifier can be considered fairly powerful.

It is important to distinguish between human interaction in classifier design and human interaction in the operation of the classifier. The former is permissible since the classifier is fixed once it has been effectively designed and trained. No further human assistance is allowed and the classifier is applied in an automatic fashion to the test set.

3.9.3 Detailed classification description

The objects to be classified in this study are connected regions of an input picture, extracted by thresholding the image. More than one threshold may have been used on any given picture, so the regions need not be disjoint; rather one may be entirely contained in another. For each region, a feature vector containing information about shape and brightness (as described in Section 3.8) is used as the sole source of information about the region for classification. The extraction procedure has somewhat preselected these regions, so that every region examined has at least minimal (20%) correspondence between its perimeter and the high-edge points, has at least minimal contrast (.2 gray level), and is of roughly appropriate size (between 20 and 1000 pixels).

3.9.3.1 Stage 1: pre-classification

If the classification is thought of as a two-stage process (shown schematically as Figure 3.9.2), the first stage is a crude "semantic" classifier which identifies some regions as having properties which indicate that they are not targets. Thus, all targets have similar height and width, seen at any aspect angle. Any region with h/w greater than 3 or less than 1/3, then, may be confidently rejected from further consideration. Similarly, targets "should" show some minimal contrast at their perimeters, a good edge-perimeter overlap, and small targets should be of nearly uniform brightness. All these criteria are set by establishing numerical thresholds such that at least 95% of the sample targets satisfy the criteria.

This is called "semantic" classification, rather than a very crude statistical classification, because the particular criteria used have been chosen to distinguish the targets on the basis of physical characteristics of true target images. A statistical classifier, even if it arrived at the same scheme, would be assessing discriminatory ability on the sample of classified regions provided for training, and could reflect any peculiarities which happened to distinguish the categories in that sample. (In the NVL data, APC's often exhibit an asymmetry which is due to the fact that most of those in the sample appear in only a single aspect. An apparently good statistical classifier

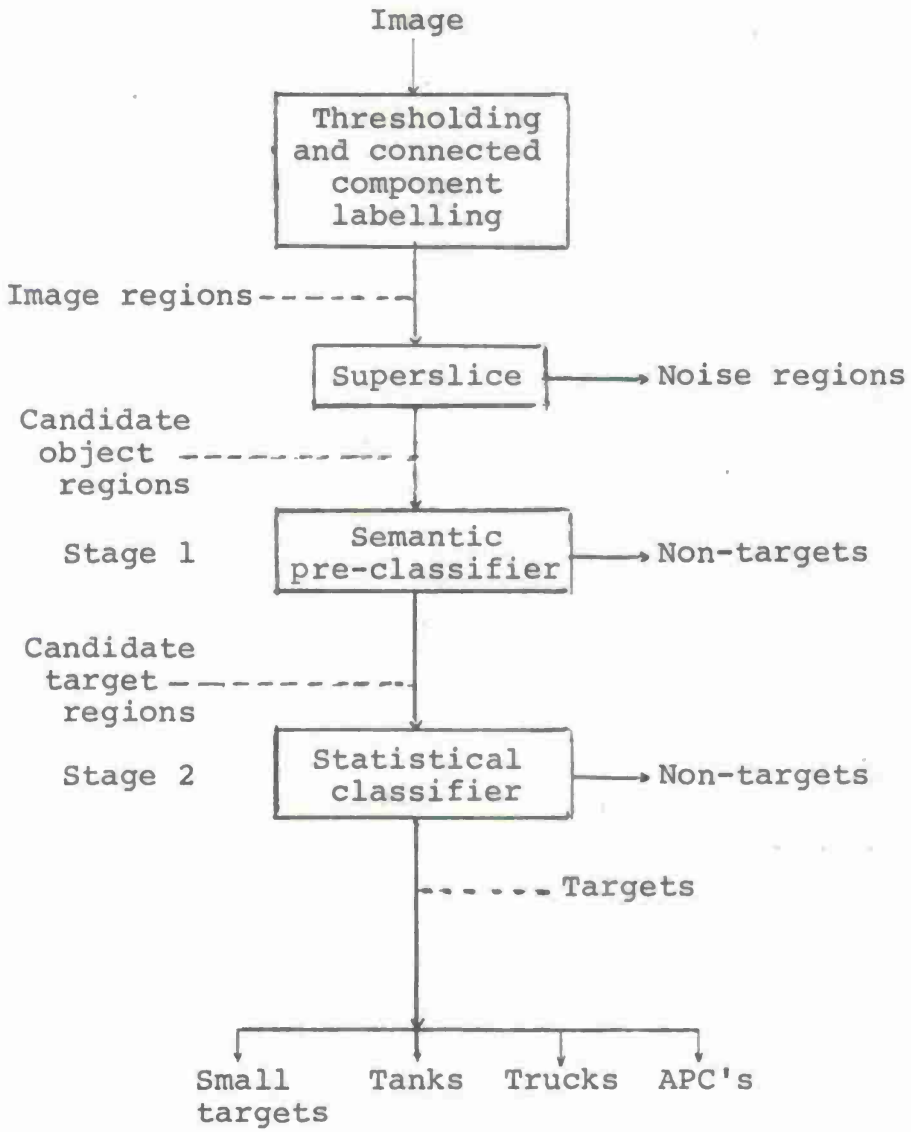


Figure 3.9.2a. The classification process.

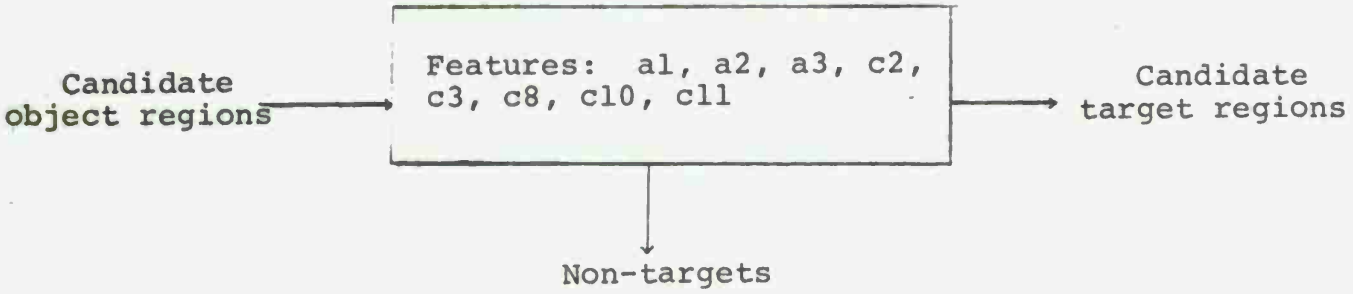


Figure 3.9.2b. Stage 1 - the pre-classifier (for feature list, see Table 3.8.1).

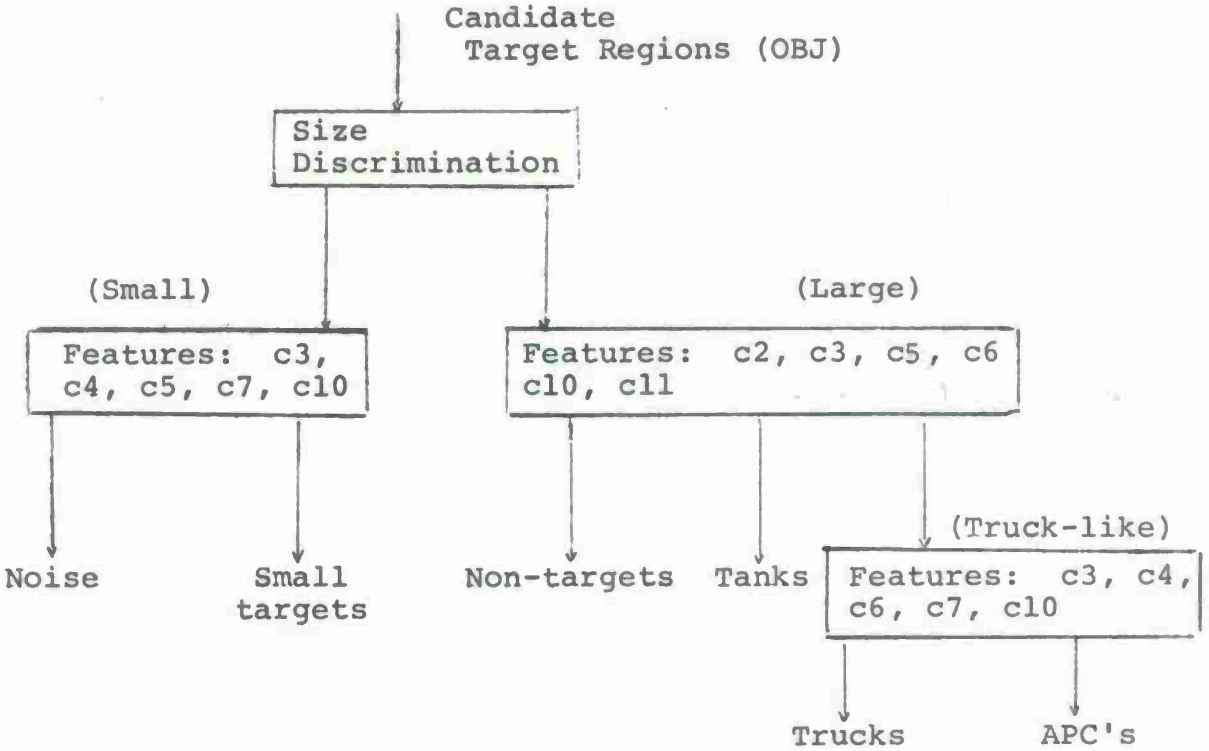


Figure 3.9.2c. Stage 2 - the classifier (for feature list, see Table 3.8.1).

could be formed which would unhesitatingly identify any APC in some other aspect as a tank.)

This pre-classification examines individual features to determine whether they could be reasonably associated with true targets, and discards "ridiculous" cases. A side-effect of this sorting is to assure that feature values seen by the subsequent statistical classifier are never very far from their characteristic values. This makes the classifier much better-behaved than one which accepts non-normally distributed features (as most do) that have not been "critiqued."

3.9.3.2 Stage 2: statistical classification

Once the set of extracted regions has been reduced to a set of bright, compact, reasonably uniform regions, statistical classification is used to assign a class to each particular combination of features (or rather, to its associated region). A great many kinds of statistical decision rules exist. Access to the MIPACS [25] interactive system allowed us to design a decision tree (each node of which is a standard classifier) for efficient classification. The system allows individual decision functions to be either linear (e.g., Fisher), quadratic, or maximum likelihood, and provided a convenient mechanism for selecting which decisions to make, and just which features to use at each decision point.

The basic structure selected was shown in Figure 3.9.2c. The first node actually represents a non-statistical selection. Because of the wide range of apparent sizes of the target images (from 25 to 1000 pixels) and the consequent wide range in visible complexity of detail, it was quickly determined that statistical classifiers would not provide good discrimination over the entire size range. (Almost every feature measured showed substantial correlation with apparent size, and since the various sample classes happened to have rather different image size distributions, our earliest classifiers used that factor as a main classification indicator.) Therefore, the first step in the

classification is a simple split on image area -- with all regions of less than 95 pixels going to the "small" subtree, and the remainder passing into the "large" subtrees. For several reasons, principally a presumed lesser urgency for detailed identification of small or distant objects and the fact that in the smallest images no significant differences between the various target classes are apparent, the small regions are simply sent to a node which classifies them as (small) "target" or "non-target" -- the specific type of target is left unspecified. For the large regions, a two-stage process followed. As neither APC's nor trucks are particularly well characterized by the features used and their distributions are very similar, they were merged into a composite "truck-like" class. Any region found to be in this class is then assigned as APC or truck by a Fisher discriminant. (A major reason for this breakdown is that it permits fairly large samples to be used at an important decision point and relegates use of the sparsely sampled truck class to a relatively inconsequential discrimination.) The principal decision was therefore between the "tank" and "truck-like" classes and the "non-target" class. Two different approaches were tried for making this decision, both based on a quadratic maximum-likelihood discriminant. These are described more fully in Section 3.9.4. One approach ("fixed classes") applied the maximum likelihood criteria directly to the tank, truck-like, and non-target

classes. The second approach included two "reject" possibilities as well -- non-target, and unclassified target. (Notice that the non-target label is applied either if a region looks sufficiently like a "typical" non-target or if the best label implies too unlikely a value for the features measured.) The latter approach was included to further minimize reliance on characterizing non-targets in detail.

Given the tree structure for the classification, the kind of classifier and the set of features at each node were determined. The number of features which can reliably be used depends on the size of the sample set used for training. Assuming that the features are chosen so as to avoid apparent vagaries in the set of exemplars, one can confidently use an additional feature for each ten samples in the smallest group, and sometimes may use up to one-third the sample number (for a linear classifier). As quadratic classifiers utilize more detail of the presumed distribution one is restricted to the conservative end of that range. These rules of thumb, while not universally valid, are nontheless useful guides.

By merging the truck and APC classes, we allow comfortable use of a quadratic classifier on five or six features at the main decision node, while the smaller samples make a linear classifier or a three or four feature quadratic

classifier more reasonable at the lower node. The "small" node could utilize five or six features -- but one is hard-pressed to find even that many which provide any discriminatory power at all. (However, one feature, E&P, is very powerful indeed.)

3.9.4 Experimental results

3.9.4.1 Feature selection

As in any classification problem, much of the initial feature selection for the vehicle recognition task was carried out informally. This phase is largely introspective, determining characteristics of the images that seem helpful for human judgement, then identifying some features that should suitably reflect these characteristics. This initial feature set (conveying "shape" and "relative brightness") is listed in Table 3.8.1, Section 3.8. All of these features seem appropriate for use with linear or quadratic classifiers.

The features were examined in several ways. First, histograms for each feature were produced for every sample class. These histograms were examined to see whether the sample distributions satisfied the criteria noted in the last section. The differentiation that appeared was interpreted as to whether it was a true difference between classes, or simply a sampling anomaly. (At this stage too, particular features might be replaced by similar features of slightly different functional form, to better satisfy the requirements of automatic classification.) Second, those features that seemed to have some merit were ranked for classification power at each node of the decision tree. The "Automask" method, available within MIPACS, was used ([25]). Briefly, Automask finds, for each feature, its "share" of the total dispersion both between and within sets,

and finds the single feature which produced the greatest comparative variance between sets. This feature is then deleted from consideration, and the other features reexamined to find the next best feature, and so on. The relative merits of the features for each node are shown below.

Node	Good features	Usable features
Small	E&P	(h/w)',(h*w)/A,(h+w)/P,diff, skewness, asymmetry
Large	E&P, diff	(h/w)',(h*w)/A,skewness, asymmetry, E_p
Trucklike	E_p, asymmetry	(h/w)',(h+w)/P,skewness,E&P

Shape features:

In the first stage, the (h/w)' height-to-width feature was useful in identifying small bright streaks as non-targets. In the statistical classifier for small targets, shape features were individually very weak in distinguishing targets from non-targets. For large targets, diff was the best shape feature at node LARGE; all the others but asymmetry were also of some use. At node TRUCK-LIKE, on the other hand, asymmetry was the best shape feature, with the remainder of no value.

Brightness-related features:

Edge-border coincidence (E&P) was by far the strongest single feature for both nodes involving target/non-target discrimination (OBJ and LARGE). For small targets, it provides nearly all the discrimination in the second stage.

For large targets, it provides evidence which is well complemented by shape information -- both must be included for adequate performance. Also very useful, particularly at stage 1, is E_p, which provides substantially different information from E&P. Gray level variance is used to some effect in the first classifier stage, but is not effective in the second stage. Perimeter contrast information appears to be much more effectively conveyed through E_p than dgl.

These rankings, while not dependable when taken alone, have been very helpful in suggesting which features could usefully be included in decisions at each node and which should be omitted. This was especially helpful in the case of the shape features, for which estimates of relative merit were not obtainable.

The final stage of feature testing was experimental. Features suggested either by Automask or by the problem definition were included in decision functions, and self-classification attempted. In many cases, the results were not satisfactory and one or more features were added or deleted until "good" results were obtained. If too many features were present in this classifier, features were removed until the best classification obtained with an acceptable number of features was found.

3.9.4.2 Classification

The NVL data base as windowed for classification purposes consists of:

75 Tanks

34 Trucks

55 APC's

164 Target windows

10 Non-target windows

174 Total windows

Associated with each window was a liberal threshold range extending from the shoulder of the background peak gray level to the highest gray level at which there was significant sensor response. Although these ranges were manually selected, this is not a significant interference with the automatic nature of the algorithm since the gray level ranges can be chosen by a simple scheme which identifies the background peak and proposes every threshold above the peak. (If a coarse temperature calibration is available, this task is even simpler.) See Section 3.9.4.3 for further discussion.

The Superslice algorithm was run on these windows using the selected gray level ranges. Connected components whose contrast, edge-perimeter match score and size were within tolerance were retained. The resulting sets of regions are described by the containment forests in

Table 3.9.1. Within each containment tree, Superslice selects the best exemplar(s) for the candidate object region based on edge match. Thus, every tree has one or more best exemplars associated with it. All other (non-exemplar) regions are suppressed since the algorithm has proposed better representatives for classification.

Each containment tree is manually labelled as either "target-related" (containing regions associated with the target) or noise (spatially apart from a target region) so that false dismissals can be determined.

Of the 164 target windows, two windows (64T, 86A) had containment forests with no target-related regions present. At this stage, the false dismissal rate is 2/164 ∿ 1% for Superslice. Determination of a false alarm rate is inappropriate since the discrimination performed by Superslice is "object vs. non-object," not "target vs. non-target," and there is no ground truth for the number of objects (including targets, hot rocks, trees, etc.) in the frames.

The next stage - preclassification - performs possible-target vs. non-target screening. [For the purpose of building the screening criteria and subsequent classifier, a single exemplar per target was hand-chosen. No other target-related regions were considered; all noise regions, however, were retained.] Of the 162 target windows, the preclassifier retained 161 for a false dismissal rate of 1%. In addition, 44 noise exemplars also survived as possible targets.

Window Reference Number	Lowest Threshold	Containment Forests
1T	23	X(N,TTT(PPPPPP,PP),NNN,N(N,NN),NN);NN
2T	23	TTTTTTTT
3T	25	TTTTTTT(PP,P);NN;NN
4T	30	TTTTTT
6T	25	TTTTTT
8T	26	TTTTT
9T	24	TTTTTT(P,P)
10T	25	TTTTTT
11T	25	TTTTTTTT;NN
12T	22	X(PPPPP(P,P(P,P)),N)
13T	20	XX(N,TTTT);N
14T	22	TTTTTTT
15T	30	TTTTT
16T	24	TTTTTTT
17T	26	TTTTT
21T	26	TTTTT
22T	25	TTTTTT
24T	29	TTTTT
26T	26	TTTTT
28T	27	TTT
31T	27	TTT
32T	21	X(TTTT,N,N)
33T	23	VTTTT;N
34T	26	TTT
35T	24	TTTT;N
38T	24	TTTTTTT
40T	23	TT;NN;N
42T	24	TTTTT(P,P(PP,PP))

Table 3.9.1. Containment forests of regions extracted by Superslice(Tanks). "AB" means that region A contains region B. "A(B,C)" means that region A contains the disjoint regions B and C. "A;B" means that A and B are disjoint regions in the window. Underlined letters denote "best" exemplars of the target region. Target trees begin at lowest threshold.

Legend:
- T target
- P partial target
- X target with additional noise
- O target invisible in noise
- N noise region
- F fiducial mark
- V target region not present at this threshold

43T	26	TTTTT
45T	25	TTTTTT
46T	26	TTTTTT
48T	24	TTTTTTTTT
50T	22	OTTTT
51T	24	TTTTT
52T	23	TTTTT
53T	23	TTTTT
54T	23	TTTTT
55T	23	TTTTT
56T	22	T;N;N
57T	22	TTT;NN;N
58T	20	X(NN,NN,TT)
59T	21	X(TT,NN);N;N
61T	43	TTTTTT
62T	24	TTTT;N
63T	24	TTT;N
64T	28	FFFFF;N;N (no target region found)
65T	46	T
66T	47	TT
68T	26	TTTT;N
69T	26	TTTT;NN
73T	43	TTTTTTT
74T	45	TTTTT
75T	22	TTTTTT(P(P,P),P)
76T	23	TTTTTTTTTTTT
78T	27	TTTTTTT(P,P)
79T	24	TTT(PPPP,PPP)
80T	22	TTTT(P,PPPPP(P(P,P),PP))
89T	23	TTTTTTTTTT
92T	22	TTTTTTTTTTT
95T	24	TTTTTTTTTT
99T	21	TTTTTTTTTTTT
105T	24	TTTTTTTTTT(P,P,P)
109T	28	TT(PPPP,PPPPP,P(PPPP,PP))
110T	24	TTTTTTT
114T	25	TTTTTTTTTT
122T	20	TTTTT
123T	22	TTTTTTT
124T	21	TTTTTTTT
125T	24	TTTTT
126T	23	TTTT
127T	24	TTTTTTTT
128T	23	TTTTTTT
129T	24	TTTTTTTT
130T	25	TTTTTT
131T	26	TTTTT

Window Reference Number	Lowest Threshold	Containment Forests
3R	23	X(TTT,NNN(NNNN,NNNNN));N
4R	22	TTTTTT;N;N;NNN
6R	23	OTTTTTT;NN
9R	23	X(TTTT,N(NN,N),NN,N,N)
18R	26	VTTT;N
22R	24	TTTTTT
24R	28	X(X(TTT(PP,P),N))
26R	27	TTT;N
31R	26	OTTTT
32R	21	X(PP,N,N,N);NN
33R	23	X(X(TTT,N),N
34R	24	VVTTT;N;N;N
35R	23	TTT;N;N;N
41R	25	TTTTTTTTTT
47R	25	TTTTTTTTTTT
51R	25	TTT;N;N
52R	23	TTT
53R	24	TT
54R	23	TT;N
55R	23	VTTT;N;N
56R	24	TTTTTT;NNN
57R	24	TTTT;NNN;N;NN
58R	24	TTTT;NN
59R	23	TTTT;NN;N;N
71R	44	TTTTT
72R	46	TTT;NN;N
77R	27	TTTTTT(P,P)
100R	23	TTTTTTT
104R	27	TTTTTTT
109R	27	TTT(P,PPP)
123R	27	X(TT(P,P),N)
133R	27	TTTT
134R	27	XTTT(P,P)
135R	26	TTTT

Table 3.9.1. Continued. (Truck windows)

Window Reference Number	Lowest Threshold	Containment Forests	
21A	26	TTTTTT	
22A	22	TTTTTT	
24A	28	TTTTT	
27A	27	VTTT;N	
32A	25	TT	
33A	25	T;N;N	
34A	26	TTT	
35A	25	TT	
37A	27	TTTTTTTT	
38A	23	TTTTT	
42A	24	TTTT(PP,PP)	
44A	28	TTTTTTTT	
45A	26	TTTT;N;N	
46A	26	TTTTTT	
48A	26	TTTTTTT	
50A	24	TT	
51A	25	TTTT;N;N	
52A	25	TT;N;N	
53A	24	TTT;N;N;NN	
54A	25	TTT	
55A	26	T	
56A	25	TTTT	
57A	24	TTTT	
58A	25	TTTTT	
59A	24	TTTTTTT	
61A	41	TTTTTTT	
73A	43	TTTTTT	
74A	43	TTTTTTT	
75A	25	TTTTTTT;N	
76A	26	TTTTTTTTT	
78A	31	P(PP,P)	
79A	25	TTT(PPPP,PPP)	
80A	24	TTTTTTT	
86A	24	FFFFF;NN;N;N	(no target related region found)
90A	25	TTTTTTTTTT	
91A	26	TTTTTTTTT	
93A	26	TTTTTTTT(P,P)	
94A	26	TTTTTTTT	
96A	27	TTTTTT	
97A	24	TTTTTTTT;N	
98A	24	TTTTT	

Table 3.9.1. Continued. (APC windows)

```
101A        44      TTTTT
102A        44      TTTTT;N
111A        24      TTTTTTTTT
112A        24      TTTTTTTTT
113A        23      TTTTTTTT
114A        29      X(TT(P,P),NN)
115A        24      TTTTTTTTT(P,P)
122A        23      TTT
123A        24      TTTTTT
125A        24      TTTT
127A        24      TTTT
129A        26      TTTT
130A        23      TTTTTT
```

The false dismissal was 66T (small, very faint).

After preclassification, 150 selected target exemplars and all 44 noise exemplars were split into a training set (74 targets and 22 noise regions) and a test set (76 targets and 22 noise regions). The training set was used to design the optimum decision rule. It was felt that similar results in classifying both sets would then indicate that the classifier had utilized robust characteristics of the target class and thus could be expected to give similar results on further data of the same type.

A linear discriminant was used at the trucklike node while a maximum likelihood discriminant was used at the small target/non-target node. Five features were used at both nodes, of which four were the same: (h*w)/A, (h+w)/P, asymmetry, E&P. The fifth feature was diff for the small target discriminant and skewness for the truck/APC discriminant. The large targets are divided into three classes (tank, truck/APC, other) by a quadratic maximum likelihood discriminant using six features: (h/w)', (h*w)/A, diff, skewness, E&P and E_p. Two different procedures for classifying large regions (> 94 pixels) were tested. One procedure attempted to discriminate between four fixed classes (tank, APC, truck, other); the other procedure used three classes (tank, APC, truck) and two "reject" categories (non-target, unidentified target). Both used identical polynomial maps into decision space. In the latter classifier, however, the maximum likelihood class assignment of a region had to be significantly better than for random noise regions (otherwise, the

non-target class was assigned) and significantly better than the next best target class assignment (otherwise, it was called an unidentified target).

The detection results using the fixed class classifier on the 150 selected target exemplars are summarized by:

	Train	Test	Total
Large	53/53	53/55	106/108
Small	20/21	20/21	40/42
Total	73/74	73/76	146/150

where "M/N" means "M successes out of N tries." This classifier thus appeared to be robust.

Table 3.9.2 displays the results of this classifier for all extracted regions, including all target and noise exemplars. A false dismissal for a window containing a target occurs when no target exemplar (at any of the thresholds) is classified as a target (i.e., classified as tank, truck or APC). Similarly, a false alarm is any noise exemplar (i.e., not associated spatially with a target region) classified as a target. However, multiple exemplars for the same noise region are counted only once. In effect, we are counting the image regions (as opposed to exemplars) which are classified as target regions by at least one exemplar. If a region is, in fact, a target re-

	Thresholds																												
Frame	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1T				[0	0	+	+	+	+	+	0	0	0]																
2T				[+	+	+	+	+	+	+	+]																		
3T						[0	0	0	0	±	±	+	+	+]															
4T											[+	+	+	+	0	0]													
6T						[+	+	+	+	+	+]																		
8T							[+	+	+	+	+]																		
9T					[0	0	0	0	+	+	0]																		
10T						[0	0	0	0	+	+]																		
11T						[+	±	+	+	+	+	+	+]																
12T			[0	+	0	+	0	0	0	0]																			
13T	[0	0	+	+	+	0]																							
14T			[+	+	+	+	+	+	+]																				
15T											[+	+	+	+	+]														
16T					[+	+	+	+	+	+	+]																		
17T							[+	+	+	+	+]																		
21T							[+	+	+	+	+]																		
22T						[+	+	+	+	0	+]																		

Table 3.9.2 Region classification (tank windows).

Each entry represents the outcome of the classifier for the purpose of target detection. Brackets indicate the range of thresholds considered for each window. "+" means that the target was detected at that threshold. "0" means that the target was dismissed. "-" indicates a false alarm for that threshold. No window had two or more distinct false alarm regions.

Frame	Thresholds																													
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
24T										[+	+	+	+	+]															
26T							[0	+	+	+	0]																		
28T							[	+	+	+	]																		
31T							[	+	+	+	]																		
32T	[		0	0	+	+	0	]																					
33T				[	+	+	+	+	]																				
34T						[	+	+	+]																				
35T					[0	+	+	+]																					
38T					[0	0	0	+	+	0	0	]																	
40T			[	+	+			]																					
42T					[+	+	0	0	+	+	0	0	]																
43T							[+	0	+	+	+]																		
45T						[+	0	+	+	+	+]																		
46T							[0	0	+	+	+]																		
48T					[+	+	+	0	+	+	0	+	0	]															
50T			[+	+	+	+	0]																						
51T					[+	+	+	+	+]																				
52T				[+	+	+	+	0]																					
53T				[+	+	+	+	+]																					
54T				[+	+	+	+	+]																					
55T				[+	+	+	+	+]																					
56T			[+		]																								
57T			[+	+	+		]																						
58T	[0	+	+			]																							
59T		[0	+	+		]																							
61T																								[+	+	+	+	+	+
62T					[+	0	+	+]																					
63T					[+	+	+]																						

Frame	Thresholds																													
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
64T									[	0	0	+	0	+	+]														
65T																											[+]		
66T																												[0	0]
68T							[+	+	+	0]																			
69T							[+	+	+	0]																			
73T																								[+	+	+	+	+	+]
74T																										[+	+	+	+]
75T			[+	+	+	+	+	+	+	+]																			
76T				[0	+	+	+	+	+	+	+	+	+	+	0]														
78T								[+	+	+	+	+	0	+	0]														
79T					[+	+	+	+	+	0	+]																		
80T			[+	+	+	+	+	+	+	+	+	+	+]																
89T				[+	+	+	+	+	+	+	+	+	+]																
92T			[+	+	+	+	+	+	+	+	+	+	+	0]															
95T					[+	+	+	+	+	+	+	+	+	+]															
99T		[0	0	+	+	+	+	+	+	+	+	+	0]																
105T					[+	+	+	+	+	+	+	+	+	0	0]														
109T									[+	+	0	0	0	0	0]														
110T					[0	+	+	+	+	+	+]																		
114T						[+	+	+	+	+	+	+	+	+	0]														
122T	[+	0	+	+	0]																								
123T			[+	+	+	+	+	+	0]																				
124T		[0	+	+	+	+	+	0	0]																				
125T					[+	+	+	+	0]																				
126T				[0	+	+	+	+]																					
127T					[+	+	+	+	+	+	+	0]																	

	Thresholds																														
Frame	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
128T				[	+	+	+	+	0	0]																			
129T				[	+	+	+	+	+	+	+	0]																	
130T						[+	+	+	+	+	0]																		
131T							[0	+	+	+	0]																		

Frame	Thresholds																												
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
3R				[+	+	±	0				–	–			]														
4R			[+	+	+	0	0	0		]																			
6R				[+	+	+	+	0	+	+]																			
9R				[0	+	+	0	+		]																			
18R							[	+	+	0	]																		
22R					[0	0	+	0	+	0]																			
24R									[+	+	+	0	+	0	0]														
26R								[	0	+	0	]																	
31R							[	+	+	+	0]																		
32R		[0	0	0		]																							
33R				[	0	0	+	+]																					
34R					[		0	+	+]																				
35R				[0	0	0]																							
41R						[0	+	0	+	+	+	0	0	0	0	0	]												
47R						[+	+	+	+	+	+	+	+	+	0	0	]												
51R						[0	+	0		]																			
52R				[0	0	+		]																					
53R					[0	+	]																						
54R				[0	+	;																							
55R				[	0	+	0]																						
56R					[+	+	0	0	0	+]																			

Table 3.9.2 (continued): Truck windows.

	Thresholds																														
Frame	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48		
57R					[+	0	+	+		]																					
58R					[+	+	+	+		]																					
59R				[0	+	+	0		]																						
71R																									[+	+	+	+	0]		
72R																											[+	+	0]		
77R								[+	+	+	+	+	+	0]																	
100R				[0	+	+	+	+	0	+]																					
104R								[+	+	+	+	+	+	+]																	
109R								[0	+	+	+	+	0]																		
132R								[+	+	+	0]																				
133R								[+	+	+	0]																				
134R								[+	+	+	0]																				
135R							[+	+	+	+]																					

Frame	Thresholds																												
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
21A							[+	+	+	+	0	0]																	
22A			[+	+	+	+	0	0]																					
24A									[+	+	+	+	+		]														
27A							[0	+	+	+	]																		
28A								[	+	+	0]																		
32A						[+	+		]																				
33A						[0			]																				
34A							[0	+	+		]																		
35A						[0	+			]																			
37A								[+	+	+	+	+	+	+	+	]													
38A				[+	+	+	+	0	]																				
42A					[+	+	+	+	+	0	]																		
44A									[+	0	+	+	+	0	+	0		]											
45A							[+	+	+	+	0		]																
46A							[0	+	+	+	+	0	]																
48A							[+	+	+	+	+	+	0]																
50A					[+	0		]																					
51A						[0	+	+	0	]																			
52A						[0	+			]																			
53A					[+	+	+]																						
54A						[+	0	0	]																				
55A							[+			]																			
56A							[+	0	+	0	]																		

Table 3.9.2 (continued): APC windows.

Frame	Thresholds 20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
57A					[+	0	0	0]																					
58A						[+	0	+	+	+]																			
59A					[0	+	+	+	+	+	0]																		
61A																						[+	+	+	+	+	+	+]	
73A																								[+	+	+	+	+	+]
74A																								[+	+	+	+	+	+]
75A						[+	+	+	+	0	0	0]																	
76A							[+	+	+	+	+	+	+	+	+]														
78A												[+	0	0]															
79A						[0	+	+	+	+	0	+]																	
80A					[+	+	+	+	+	+	+]																		
86A					[_	_			]																				
90A						[+	+	+	+	+	+	+	+	0	0]														
91A							[+	+	+	+	+	+	0	+	+]														
93A							[+	+	+	+	+	+	+	+	0]														
94A							[0	+	+	+	+	+	+	+]															
96A								[+	+	+	+	+	+]																
97A					[+	+	+	+	+	+	+	+]																	
98A					[+	+	+	+	+]																				
101A																									[+	+	+	+	+]
102A																									[0	+	+	+	+]
111A				[+	+	+	+	+	+	+	+	0]																	
112A					[+	+	+	+	+	+	+	0	0]																
113A				[+	+	+	+	+	+	+	0]																		
114A										[+	+	+	+]																
115A					[+	+	+	+	+	+	+	+	+	0]															

	Thresholds																												
Frame	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
122A				[+	+	0]																							
123A					[+	+	+	0	+	0]																			
125A					[+	+	+	+]																					
127A					[+	+	+	+	]																				
129A							[+	+	+	0]																			
130A				[+	+	+]																							

Frame	Thresholds 20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
2N								[	_			_	_	]															
8N								[				_			]														
14N	[			]																									
20N							[								]														
26N		[				]																							
32N			[			]																							
38N	[		]																										
44N			[				]																						
50N			[			]																							
56N	[				]																								

Table 3.9.2 (continued): Noise windows.

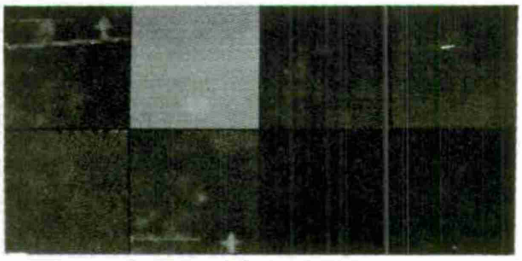
gion and some exemplar of it is called a target, that is a success. If no exemplar is so called, then a false dismissal has occurred. Finally, if the so-called target region does not, in fact, contain a target, then a false alarm has occurred.

The classifier results consist of 6 false alarms and 3 false dismissals from the 162 target windows and 2 more false alarms from 10 non-target windows. No window contained more than one false alarm cue. Details are as follows:

False Dismissals	False Alarms
32R	3T
35R	11T
33A	3R
	56R
	59R
	86A
	2N
	8N

Figure 3.9.3a displays the 6 (total) false dismissals. Masks of the 8 false alarms along with their gray level windows are shown in Figure 3.9.3b.

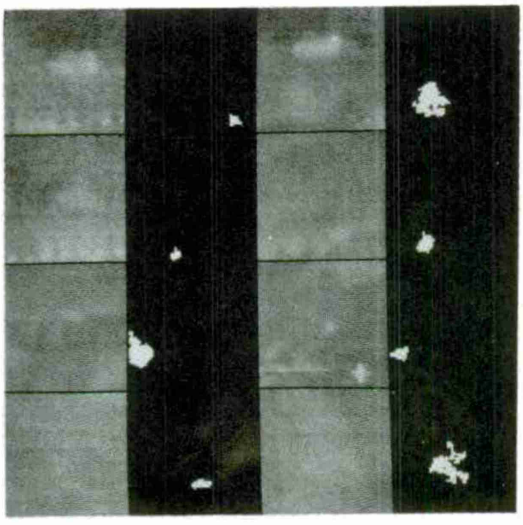
The question of how target identifications can be made in this environment of multiple exemplars, while secondary to the task of detection, is an interesting



64T 66T 32R 35R

36A 86A

a.



3T 11T

3R 56R

59R 86A

2N 8N

b.

Figure 3.9.3. Classification results for NVL data base.

a. Six false dismissals.
b. Eight false alarm region masks with their gray level windows.

one. Since each exemplar in a containment tree can be classified independently, there are many ways of arriving at a final region label. Section 3.9.5 discusses the use of context and considers the identification of object regions from the classifications in their containment trees as an example of context. We discuss the issue here simply from the point of view of critiquing the classifier performance. For each containment tree containing at least one exemplar classified as a target, we chose the target type of the exemplar with the best edge-match (E&P) score in the tree and used that target type to designate the region. In the event that the "best" exemplar was not described as a target, we labelled the object region "unknown target". Only large targets were considered, since small targets while detectable were not considered identifiable.

In a test which classified all best exemplars of large targets (55 tanks, 21 trucks, 36 APC's) the between-types confusion matrix was:

		classified as			
		T	Tr	A	UT
A priori	T	40	5	6	4
	Tr	6	8	7	0
	A	9	5	20	2

where "UT" is the "unknown-target" type. The 8 false alarms were classified as 1 truck, 2 APC's, and 5 small

targets. Between-class confusion is high, with tanks being the most successful class. Trucks and APC's were often confused with tanks. A number of reasons can be advanced for this performance. First, tanks were the most numerous target and therefore could be identified most confidently. Second, large APC's appeared with the wooden wave deflection board in view, producing a characteristic "c" shape. No attempt was made to utilize this special knowledge. Third, the large targets appeared in only a single aspect and no generalized shape descriptors separating the different types could be extracted realiably. It seems most sensible to model the target types as three-dimensional objects and to derive discriminators from their inherent shape and size differences from all aspects.

The second classifier (which applied a threshold to reduce the false-alarm rate) did not improve classification as might have been expected. Any threshold which would have reduced the number of false alarms also caused a number of false dismissals. Thus while the method might be of use, its utility could not be judged on the limited data set available especially since there is no model relating the false alarm rate to the false dismissal rate.

We may summarize the principal classification results as follows: the false dismissal rate of the system is less than 4%, giving a system detection rate of 96%.

The false alarm rate, based on the number of false alarm regions per unit area, is 8 false alarms in 174 (128x128) windows. Assuming there are 500x800 pixels per frame and that a target occupies about 1/10 of a window, we conclude that the total processed area corresponds to about 6 frames. Thus the false alarm rate is 8/6 or 1.3 per frame. A separate test of the false alarm rate was made using a set of four 512x512 pixel frames (Figure 3.9.4). All available targets were detected. In addition, 4 large false alarms and 8 small false alarms were detected (see Figure 3.9.5). However, 5 of the 8 small false alarms corresponded to fiducial marks. Moreover, one large false alarm (in F1) appears to be a target. In any case, 7 false alarms in 4 frames agrees well with the previous estimate of the false alarm rate.

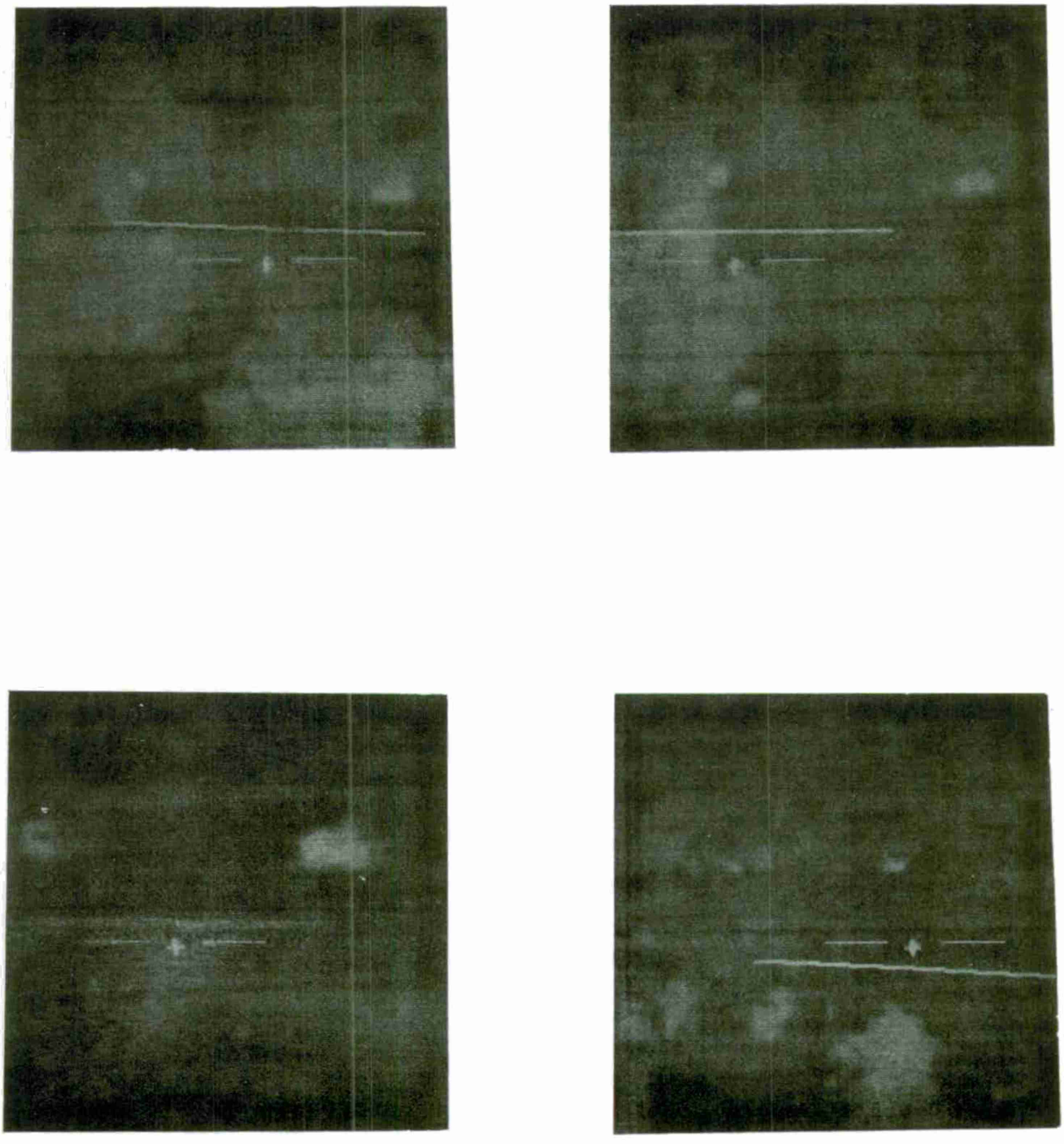


Figure 3.9.4. Four 256x256 frames (after median filtering and sampling).

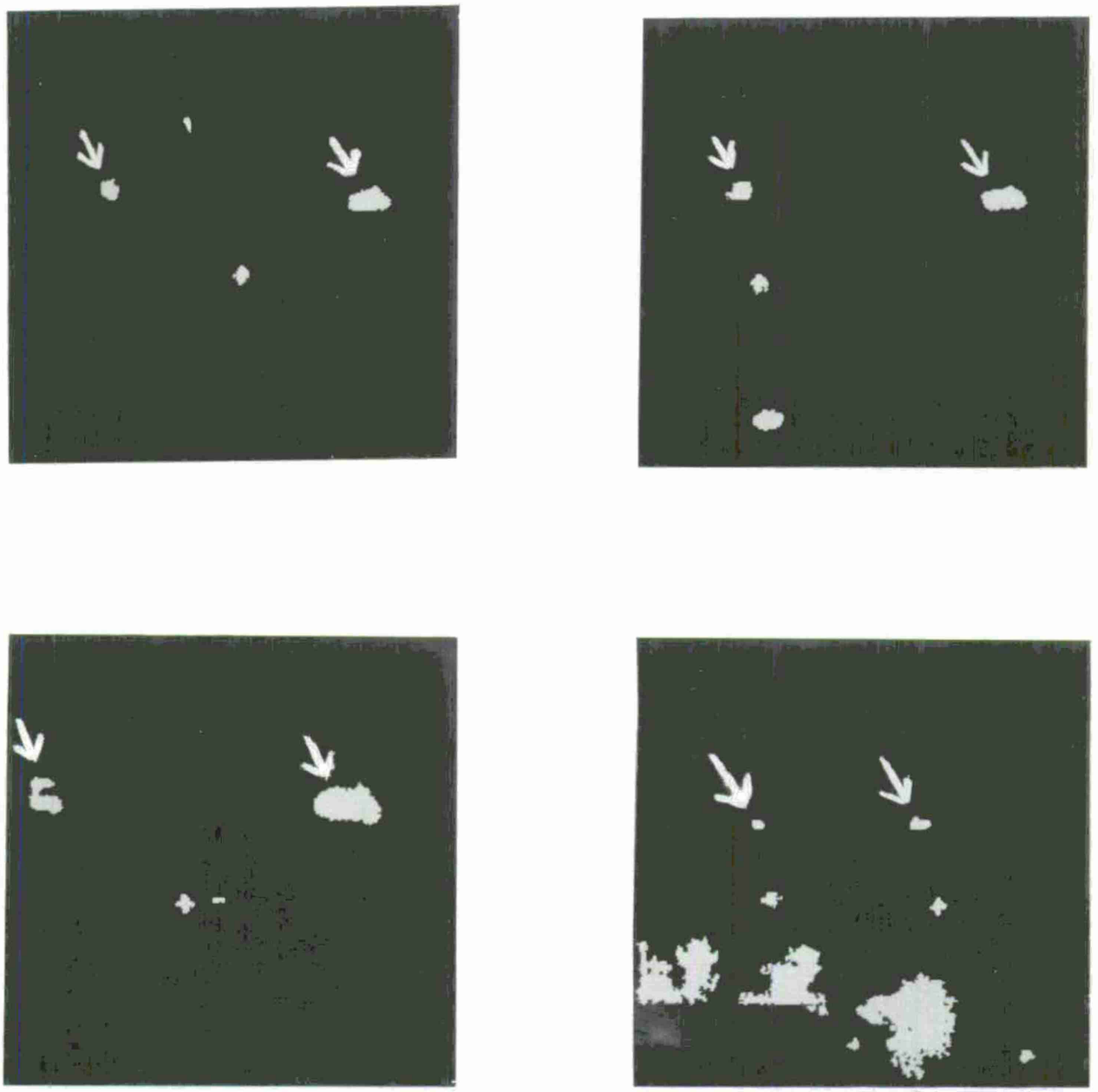


Figure 3.9.5. Cued regions in the four frames of Figure 3.9.4. All targets were detected (masks indicated with arrows), along with 12 false alarms (5 corresponding to fiducial marks).

3.9.4.3 Threshold selection evaluation

Our method of threshold range selection was described previously. However, it bears repetition in this section. Using the histogram of gray levels (perhaps of the previous image), choose as a range the sequence of gray levels from the mode to the highest gray level with appreciable response (e.g., more than 5 points). The previous subsection demonstrated that this brute force approach gave excellent system detection efficiency. Naturally, the liberal range of thresholds has important effects on system architecture, as discussed in Section 4.

Since the number of thresholds used determines the time cost (in a sequential implementation) or the hardware replication cost (in a parallel implementation), it is appropriate to consider methods which can accommodate a limited number of thresholds. "Intelligent" methods of threshold selection are discussed in Sections 3.5 and 3.10.1. We wish to consider "brute force" methods which select thresholds at every other gray level, at every third gray level, etc.

As may be seen from Table 3.9.2, correct target detections for single windows tend to occur in extended runs. Table 3.9.3 provides a histogram of run lengths. In general, large targets had better contrast and their detections were stable over long runs. Small targets were fainter and were detectable over only a few

Run length	# of windows	Cumulative count	% of 164 windows
0	5	164	100
1	17	159	97
2	25	142	87
3	29	117	71
4	27	88	54
5	19	61	37
6	12	42	26
7	10	30	18
8	8	20	12
9	7	12	7
10	3	5	3
11	2	2	1

Table 3.9.3. Statistics of longest runs of correct target detections in 164 target windows.

thresholds at most. Table 3.9.3 shows what percentage of the targets were detected within runs of I or longer for I = 1,2,... . Thus the false dismissal rate would be 11% if every other threshold in the range were omitted. Since there were so few false alarms, it is not possible to give comparable statistics of any reliability, but any scheme which considers fewer exemplars is bound to detect fewer false alarms.

From a slightly different point of view, we might consider how to allocate a fixed number of thresholds within a given gray level range. In the Hardware Technology section, the design assumes that five thresholds were implemented in parallel hardware. Thus, for a gray level range of 10, thresholds would occur at every other gray level; for a range of 20, thresholds would occur at every fourth gray level. If we use the gray level ranges indicated by brackets in Table 3.9.2 and distribute N (=1,2,3,...) thresholds equally spaced (where feasible) throughout the range, we compute the following results:

N	# False Dismissals	# False Alarms
1	25	1
2	14	3
3	7	7
4 and above	5	8

Thus, for four or more thresholds equally spaced throughout the available gray level range of each window, no

additional false dismissals occurred beyond those already dismissed using the whole range. Interestingly, for small N the increase in false dismissals is just about compensated by the decrease in false alarms. One is doubled as the other is halved.

Naturally, the threshold ranges depend both on window size and on window content. It is therefore not likely that three thresholds will be sufficient in practice. The best choice of N, the number of thresholds, will result from estimating the probability/cost tradeoff for faint targets. Given a range of x gray levels for target regions, N should be about x/2 or x/3, which for the current data base suggests that N should lie between 5 and 10. For an extension to image sequences, see Section 3.10.1.

3.9.4.4 Classifier extension

An attempt was made to apply the classifier derived from the NVL data base to a different set of thermal images. The Alabama data base is a set of imagery taken with a thermoscope. The actual sensor data are classified; radiometric noise was added to mask the source. Figure 3.9.6 exemplifies the type of imagery involved. The gray level histograms are not smooth and in some cases runs of gray level bins contain no points. Median filtering (using odd sizes) cannot be used to smooth such images since it preserves false contours. Median filtering using even sizes provides a small degree of smoothing. We elected to smooth by locally averaging over a 2x2 neighborhood just to introduce sufficient gray level variation so that 5x5 median filtering would be effective.

The resultant images were windowed and threshold ranges were selected. The Superslice algorithm was then applied in order to extract candidate object regions. It was necessary to increase the contrast threshold since the inherent contrast (including false contours) was higher than in the NVL data base. With this adjustment, the Superslice algorithm extracted regions corresponding to 64 out of 65 targets. After classification, 60 out of 65 were detected. In addition, there were 3 false alarms in the 48 64x64 windowns considered (although one of the false alarms appears to be a target missing from the ground truth).

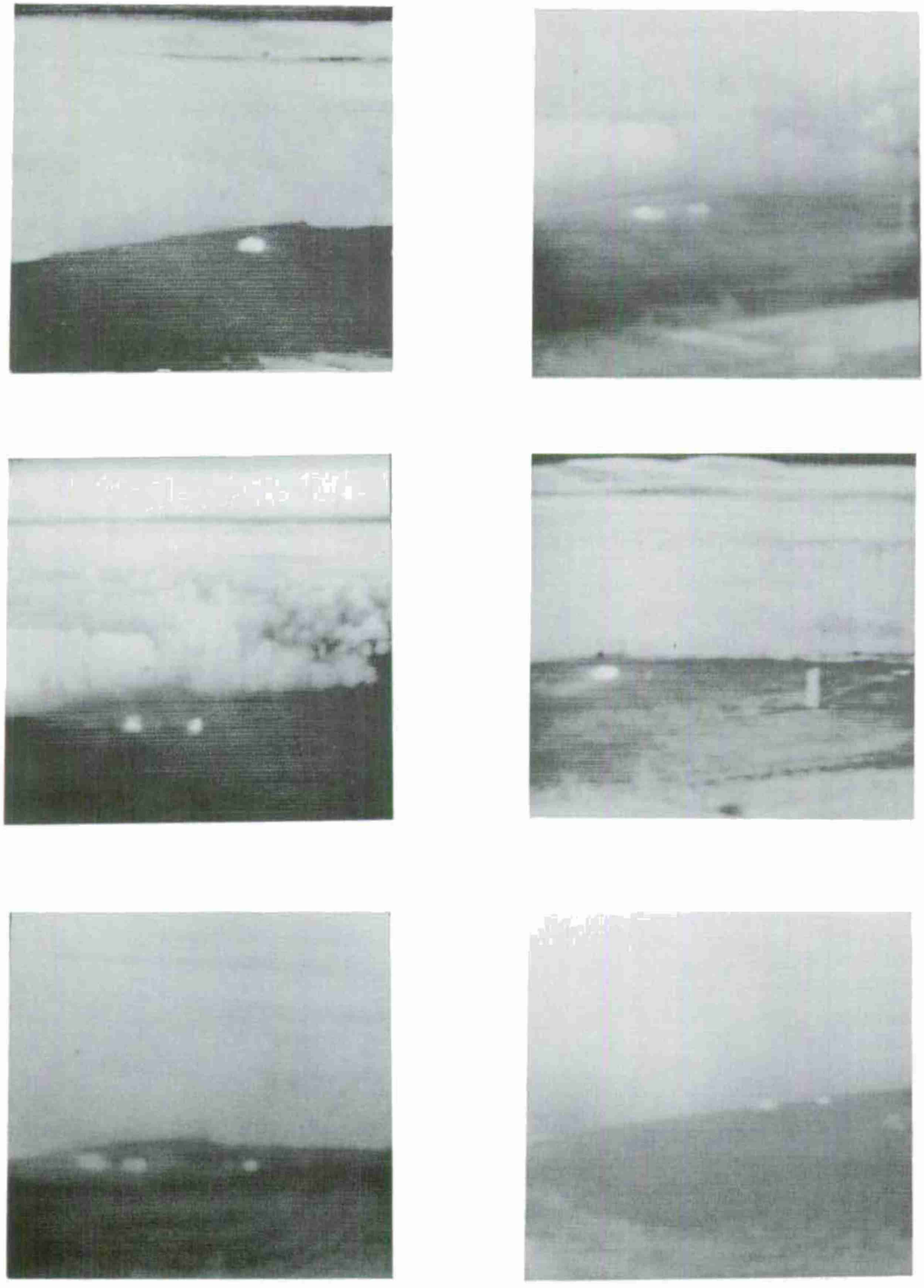


Figure 3.9.6. Alabama data base (selected frames).

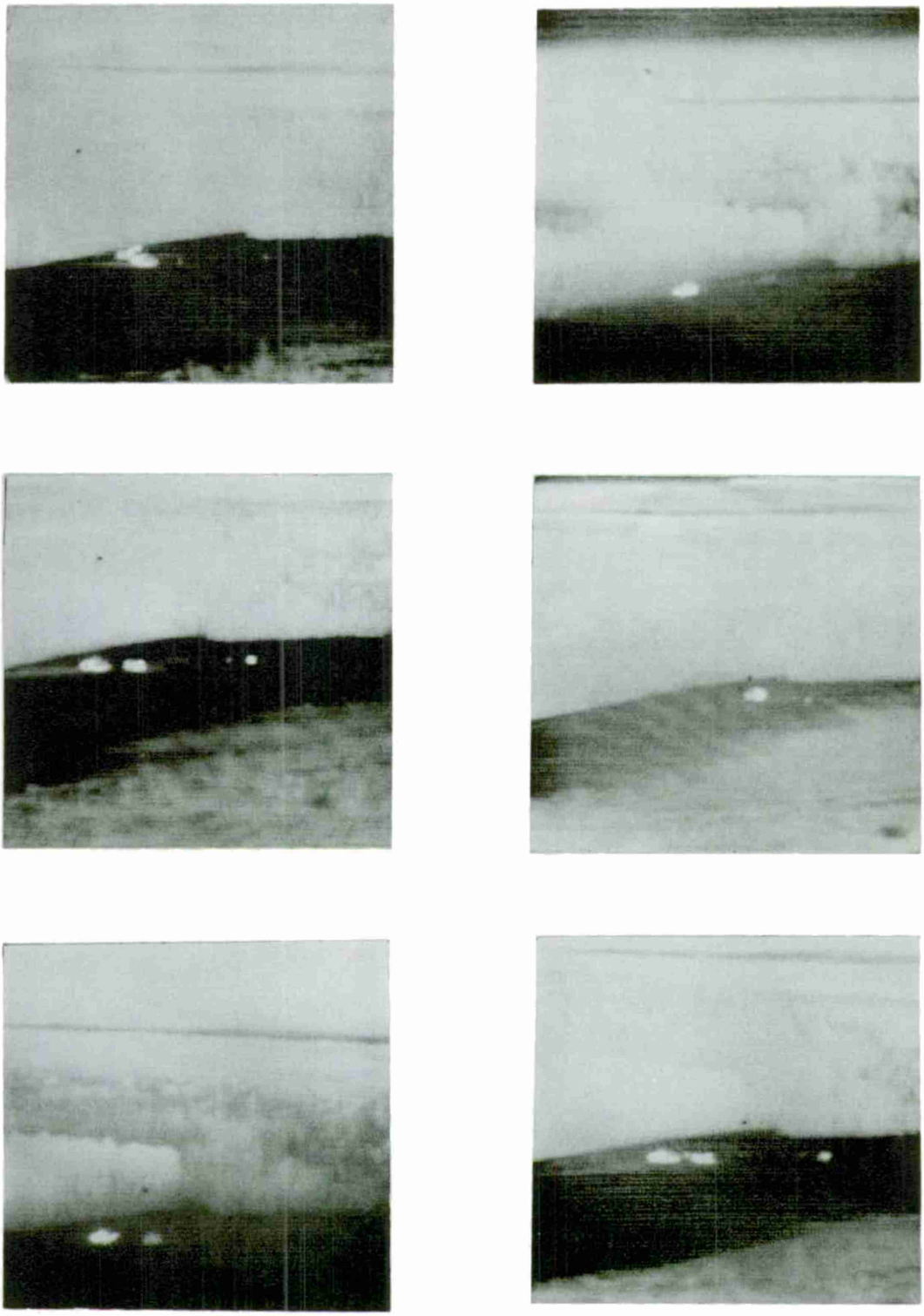


Figure 3.9.6 (continued)

3.9.4.5 Feature data base

This subsection tabulates for each window all exemplars and their associated feature vectors (Table 3.9.4). It also includes the feature weightings for each node of the classifiers and the associated thresholds (Table 3.9.5).

TANKS

WINDOW	CC#	THRESH	LOG A	H/W	(H/W)²	H*W/A	(H*W)/P	DIFF	SKEW	ASYM	SDEV	GL DIFF	E&P	EP
2T	6	25	5.65	.53	.82	1.33	.41	.60	2.16	.60	3.42	4.23	.53	3.12
3T	5	30	4.96	.52	.39	1.52	.39	.58	3.11	.56	2.28	1.74	.49	3.67
4T	4	32	4.98	.41	.62	1.42	.37	.72	2.85	.71	1.73	1.75	.50	3.15
6T	3	28	5.19	.42	.56	1.37	.45	.70	2.87	.69	4.33	5.06	.73	8.25
8T	4	27	5.42	.44	.55	1.33	.43	.98	1.92	.68	3.36	4.80	.76	6.52
9T	4	28	5.14	.42	.58	1.55	.30	.70	2.25	.59	3.15	2.17	.34	2.16
10T	4	27	6.35	.46	.50	1.43	.28	.65	1.03	.65	2.50	1.66	.39	1.67
11T	2	31	4.90	.35	.75	1.40	.45	.78	3.27	.76	4.33	2.82	.68	6.98
14T	4	25	5.16	.40	.63	1.38	.45	.72	1.91	.72	3.84	4.16	.57	4.67
15T	4	31	5.39	.42	.58	1.38	.40	.70	1.32	.70	3.32	3.09	.59	4.44
16T	5	26	6.09	.40	.64	1.31	.44	.73	1.90	.73	3.49	4.73	.61	4.54
17T	2	29	6.20	.39	.65	1.36	.46	.73	2.20	.73	3.16	4.09	.80	6.88
21T	3	28	6.50	.70	.12	1.39	.41	.34	1.36	.34	1.94	2.86	.63	2.78
22T	5	26	6.54	.61	.24	1.39	.42	.45	1.67	.45	1.86	3.04	.65	3.09
24T	3	31	6.52	.67	.16	1.37	.44	.38	1.01	.38	1.92	3.09	.65	3.51
26T	3	28	5.38	.44	.54	1.30	.43	.68	1.00	.68	1.03	1.40	.92	2.88
28T	2	28	5.40	.55	.34	1.32	.44	.54	1.90	.54	.35	1.33	.89	2.71
31T	2	28	5.54	.49	.43	1.31	.43	.61	1.28	.61	.99	1.65	.88	2.74
34T	3	26	4.56	.60	.00	1.35	.43	.22	3.42	.15	1.65	2.61	.58	2.29
38T	4	27	6.35	.53	.37	1.34	.39	.56	2.77	.55	1.12	1.29	.51	1.49
42T	4	26	6.81	.62	.23	1.36	.30	.45	1.75	.45	1.65	2.25	.43	1.59
43T	2	29	6.49	.56	.32	1.38	.31	.52	2.36	.51	1.72	2.03	.52	2.42
45T	3	28	6.60	.61	.25	1.31	.46	.46	2.83	.45	2.02	2.89	.76	4.34
46T	3	28	6.71	.58	.29	1.36	.43	.50	3.35	.47	1.62	2.47	.70	2.26
48T	4	29	6.43	.54	.36	1.35	.40	.55	2.52	.55	1.35	1.65	.49	2.04
50T	3	24	4.56	.53	.36	1.38	.43	.56	3.11	.54	1.11	1.58	.79	2.53
61T	3	46	6.78	.95	.15	1.43	.42	.05	1.47	.05	1.25	2.39	.62	3.15
65T	1	46	4.55	.72	.09	1.06	.39	.31	3.11	.28	.75	1.06	.83	2.55
73T	4	46	6.76	.70	.33	1.22	.38	.35	2.62	.33	1.36	2.62	.67	3.18
74T	4	46	6.91	.65	.19	1.18	.39	.41	3.10	.39	1.20	2.50	.77	3.51
75T	6	26	6.51	.98	.18	1.24	.26	.02	5.54	.43	1.92	2.23	.56	3.45
76T	5	30	6.32	.65	.16	1.86	.37	.40	2.39	.39	1.83	2.47	.66	4.64
78T	9	27	6.71	.57	.30	1.17	.21	.51	3.62	.47	3.50	4.09	.45	3.17
79T	8	26	6.71	.68	.14	2.16	.41	.36	4.70	.18	2.51	3.37	.58	2.62
80T	12	25	6.82	.65	.19	1.79	.34	.41	3.49	.37	2.80	4.40	.47	2.77
89T	6	27	6.65	.88	.08	1.67	.39	.13	2.31	.11	3.14	3.72	.55	3.43
92T	6	28	6.38	.83	.04	1.48	.43	.18	2.00	.17	1.98	2.94	.69	4.99
95T	7	27	6.71	.48	.46	1.02	.38	.63	2.80	.62	3.31	4.55	.58	3.96
99T	5	28	6.42	.67	.16	.77	.32	.39	7.07	.59	2.10	2.91	.68	4.95
105T	11	26	6.79	.72	.10	1.32	.41	.32	3.54	.27	3.42	5.69	.62	3.90
109T	18	28	6.41	.76	.05	1.84	.15	.27	3.74	.18	3.66	3.37	.53	3.45
110T	3	28	6.50	1.18	.35	1.29	.37	-.17	4.45	.23	2.31	2.65	.57	2.95
114T	7	28	6.60	.89	.10	1.59	.44	.11	1.56	.11	2.24	3.64	.65	3.44
122T	2	23	4.89	.97	.17	1.46	.38	.03	2.47	.08	1.15	.83	.62	2.22
123T	4	25	4.82	1.11	.29	1.31	.39	-.10	3.98	.20	1.54	2.15	.78	4.81
124T	3	26	4.66	.82	.02	1.51	.41	.19	2.78	.16	1.55	1.88	.79	5.38
125T	4	25	5.15	.80	.00	1.59	.45	.22	2.81	.19	1.62	2.12	.81	3.68
126T	3	25	5.30	.89	.10	1.43	.43	.12	2.25	.09	1.69	2.46	.84	3.85
127T	6	26	5.25	1.04	.23	1.19	.34	-.04	2.99	.12	2.18	3.06	.79	4.95
128T	4	26	4.91	.74	.08	1.34	.39	.30	4.52	.07	1.85	2.75	.78	4.20
129T	5	27	5.24	1.28	.42	1.20	.41	-.26	2.81	.21	1.70	2.57	.92	6.41
130T	5	26	5.09	.64	.20	1.76	.49	.42	.70	.42	1.88	2.68	.78	5.30
131T	3	28	6.28	.99	.14	1.26	.24	.01	5.07	.36	1.06	.92	.45	1.16
13T	3	25	4.25	.39	.64	1.59	.42	.73	2.61	.73	1.13	1.41	.86	3.07
32T	2	25	4.34	.77	.04	1.36	.45	.26	2.21	.25	.77	.89	.78	1.81
33T	2	26	4.06	.94	.14	1.31	.42	.07	.51	.07	.99	1.50	.72	2.81
40T	3	24	3.71	.71	.10	1.46	.40	.32	2.36	.32	.54	.32	.62	1.24
51T	2	27	4.08	.53	.37	1.37	.45	.56	1.55	.56	1.36	1.55	.91	3.39
52T	3	25	4.04	.43	.57	1.33	.44	.69	2.92	.68	1.43	1.81	.79	4.88
53T	3	25	4.42	.51	.40	1.32	.44	.58	2.63	.59	1.53	2.29	.93	5.27
54T	4	24	4.41	.55	.33	1.36	.44	.53	2.90	.52	2.06	2.98	.88	5.22
55T	3	25	4.35	.61	.24	1.35	.42	.46	2.13	.45	1.49	1.81	.93	4.02
57T	2	23	3.64	1.07	.26	1.32	.41	-.06	1.27	.05	1.85	2.82	.76	3.60
58T	3	22	3.30	.96	.16	1.38	.37	.05	1.44	.04	1.70	2.54	.91	3.82
59T	4	22	3.76	.62	.23	1.43	.43	.45	3.32	.42	1.28	2.10	.64	2.12
62T	1	27	3.50	.65	.19	1.19	.40	.41	4.04	.34	1.61	1.73	.86	5.09
63T	1	26	3.83	.61	.21	1.38	.41	.43	1.27	.43	2.05	2.48	.63	4.07
64T	1	34	3.76	1.78	.74	2.19	.42	-.52	5.92	.16	2.66	4.47	.85	10.54
68T	2	28	4.09	.57	.31	1.35	.45	.52	1.38	.52	1.47	2.09	.85	4.55
69T	2	28	4.14	.69	.13	1.52	.44	.36	3.10	.33	1.35	2.03	.97	4.78
1T	3	30	4.56	.37	.70	1.56	.36	.76	2.46	.75	1.54	1.72	.41	2.62
1T	4	29	4.93	.36	.74	1.66	.33	.77	1.81	.77	1.80	2.26	.42	2.51
1T	5	28	5.12	.33	.82	1.61	.31	.60	1.64	.50	2.08	2.27	.45	2.72
1T	6	28	3.04	1.37	.69	1.86	.43	-.31	4.30	.16	1.00	.45	.60	3.15
1T	7	27	5.30	.35	.76	1.55	.30	.78	2.16	.74	2.44	2.68	.46	2.71
1T	8	27	3.78	1.41	.51	1.80	.42	-.33	3.99	.24	1.27	1.36	.59	3.09
1T	9	26	5.66	.34	.76	1.56	.26	.79	1.70	.79	2.68	2.59	.45	2.63
1T	11	26	3.22	1.42	.52	1.74	.41	-.34	.98	.34	1.43	3.22	.61	2.70
1T	21	24	3.16	1.21	.37	2.23	.49	-.19	6.10	.49	1.64	3.00	.68	3.82
2T	1	30	4.68	.43	.57	1.73	.42	.69	4.52	.63	2.92	3.47	.37	3.06
2T	2	29	5.00	.40	.54	1.47	.39	.73	3.05	.72	2.82	3.26	.50	3.80
2T	3	28	5.12	.39	.66	1.43	.38	.74	2.75	.73	2.88	3.29	.48	3.37
2T	4	27	5.32	.38	.69	1.34	.41	.75	2.28	.75	3.04	3.05	.51	3.47
2T	5	26	5.44	.37	.69	1.34	.41	.75	2.34	.75	3.17	3.65	.48	3.35
2T	7	24	5.76	.33	.91	1.31	.41	.80	1.30	.80	3.58	4.62	.53	3.00
2T	8	23	5.90	.33	.81	1.31	.43	.80	1.72	.80	3.81	5.11	.47	2.59
3T	1	33	3.64	.92	.13	1.34	.41	.08	2.92	.09	2.17	3.23	.68	4.52
3T	2	32	3.87	.86	.06	1.31	.41	.15	2.51	.12	2.29	2.95	.45	3.24
3T	4	31	4.73	.41	.62	1.55	.40	.72	2.52	.71	2.18	2.09	.39	2.59
3T	7	29	5.29	.49	.45	1.46	.40	.62	2.38	.61	2.43	2.74	.44	2.79
3T	9	28	5.43	.55	.34	1.42	.39	.54	.92	.54	2.53	2.85	.37	2.34
3T	10	27	5.63	.58	.29	1.39	.39	.49	1.21	.40	2.75	3.17	.32	2.19
3T	11	26	6.01	.60	.26	1.57	.29	.47	2.00	.47	2.96	3.74	.34	1.66
3T	13	25	6.24	.58	.29	1.59	.27	.50	1.63	.50	3.08	3.79	.38	1.49
3T	14	25	3.37	1.37	.48	1.57	.39	-.30	1.96	.30	3.55	7.32	.64	4.36
4T	3	33	4.75	.37	.70	1.46	.36	.76	2.59	.75	1.47	1.50	.47	2.54
4T	5	31	5.09	.44	.54	1.41	.35	.67	2.62	.67	1.91	2.37	.43	2.72
4T	6	30	5.23	.45	.52	1.43	.38	.66	2.76	.65	2.21	2.75	.41	2.94
6T	1	30	5.00	.40	.64	1.43	.42	.73	3.02	.71	4.14	4.67	.61	7.03
6T	2	29	5.12	.41	.60	1.39	.44	.71	2.90	.70	4.24	4.85	.71	7.93
6T	4	27	5.26	.42	.59	1.35	.48	.70	2.90	.69	4.45	5.50	.71	8.06
6T	5	26	5.34	.44	.55	1.36	.47	.68	2.96	.67	4.67	6.08	.62	7.19
6T	6	25	5.42	.46	.50	1.35	.45	.65	2.57	.64	4.88	6.76	.52	5.92
8T	1	30	5.14	.40	.62	1.36	.46	.72	1.68	.72	2.48	2.73	.67	6.36
8T	2	29	5.24	.42	.59	1.34	.45	.70	2.30	.70	2.72	3.37	.75	6.90
8T	3	28	5.29	.42	.59	1.34	.45	.70	2.38	.70	2.91	3.66	.73	6.68
8T	5	26	5.52	.45	.51	1.35	.42	.66	2.14	.65	3.68	5.61	.66	5.51
9T	2	30	3.47	.76	.05	1.69	.40	.27	3.68	.19	3.77	6.93	.44	3.70
9T	3	29	5.13	.36	.75	1.64	.35	.78	3.05	.76	3.21	1.69	.36	2.50
9T	5	27	5.59	.46	.51	1.43	.28	.65	1.81	.65	3.15	2.24	.32	1.95
9T	6	26	5.84	.44	.54	1.36	.33	.67	.89	.67	3.19	2.76	.33	1.71
9T	7	25	6.02	.41	.61	1.37	.33	.71	1.64	.71	3.26	3.31	.31	1.38
10T	1	30	5.26	.41	.61	2.14	.29	.72	2.94	.70	2.61	1.34	.38	1.58
10T	2	29	5.75	.43	.57	1.77	.24	.69	1.26	.68	2.47	1.30	.31	1.26
10T	3	28	6.10	.43	.56	1.55	.22	.69	2.39	.68	2.44	1.50	.34	1.51
10T	5	26	6.49	.49	.45	1.45	.31	.61	1.00	.61	2.59	1.93	.46	1.81
10T	6	25	6.73	.55	.34	1.50	.27	.54	2.46	.53	2.79	2.51	.41	1.44
11T	1	32	4.78	.35	.75	1.56	.42	.78	3.45	.76	4.35	3.32	.53	5.75
11T	3	30	5.04	.40	.64	1.41	.47	.73	2.97	.72	4.36	3.73	.63	6.24
11T	4	29	5.14	.40	.63	1.32	.46	.73	2.79	.72	4.42	4.15	.65	6.15
11T	5	28	5.19	.40	.64	1.38	.46	.73	2.86	.72	4.49	4.64	.65	5.97
11T	6	27	5.35	.43	.56	1.35	.44	.68	2.99	.67	4.74	5.24	.52	4.85
11T	7	26	5.45	.45	.52	1.34	.44	.66	2.88	.65	4.89	5.86	.42	3.78
12T	3	28	4.63	1.33	.66	2.00	.33	-.28	4.13	.14	2.09	2.97	.63	2.54
12T	5	27	5.08	.75	.06	1.74	.26	.29	2.61	.27	2.23	2.60	.45	1.77
12T	7	25	6.12	1.22	.38	2.44	.23	-.20	1.41	.19	2.38	1.10	.59	1.91
12T	8	24	6.44	1.10	.29	2.29	.22	-.10	2.06	.08	2.45	.76	.55	1.57
12T	9	23	6.88	1.21	.38	2.03	.16	-.19	3.68	.00	2.55	.42	.47	1.18
13T	2	24	3.93	.37	.71	1.47	.47	.76	3.77	.73	.84	1.09	.73	2.70
13T	4	22	4.52	.37	.70	1.33	.41	.76	.48	.76	1.43	1.95	.74	2.45
14T	1	28	4.71	.37	.71	1.56	.38	.76	2.11	.76	3.71	4.47	.53	4.46
14T	2	27	4.87	.38	.74	1.51	.42	.77	2.10	.77	3.73	3.57	.48	4.21
14T	3	26	5.05	.37	.72	1.40	.44	.76	1.65	.76	3.79	3.79	.52	4.36
14T	5	24	5.26	.41	.61	1.35	.46	.71	1.99	.71	4.00	4.56	.54	4.37
14T	6	23	5.41	.45	.52	1.34	.42	.66	1.63	.66	4.23	5.32	.49	3.87
14T	7	22	5.53	.48	.46	1.34	.44	.63	1.87	.63	4.41	5.68	.41	2.89
15T	1	34	4.72	.34	.80	2.13	.42	.80	2.29	.79	3.44	1.97	.48	3.31
15T	2	33	5.06	.40	.64	1.54	.39	.73	2.56	.72	3.30	1.90	.53	4.04
15T	3	32	5.24	.38	.69	1.42	.44	.75	2.22	.75	3.28	2.60	.53	3.95
15T	5	30	5.55	.45	.53	1.32	.41	.67	.67	.67	3.44	3.36	.52	3.87
16T	1	30	5.66	.37	.72	1.42	.39	.76	2.52	.76	2.63	2.71	.56	4.35
16T	2	29	5.85	.37	.71	1.36	.38	.76	2.48	.76	2.87	2.94	.66	5.15
16T	3	28	5.93	.37	.70	1.34	.41	.76	2.15	.75	3.05	3.43	.63	5.03
16T	4	27	6.02	.38	.68	1.31	.45	.75	2.15	.74	3.28	4.11	.70	5.27
16T	6	25	6.19	.41	.60	1.31	.41	.71	1.79	.71	3.84	5.57	.44	3.22
16T	7	24	6.30	.43	.55	1.34	.40	.68	2.11	.68	4.20	6.41	.29	1.96
17T	1	30	6.12	.38	.67	1.39	.46	.74	2.35	.74	2.92	3.31	.69	6.07
17T	3	28	6.25	.41	.61	1.36	.46	.71	1.96	.71	3.36	4.76	.76	6.51

Table 3.9.4. Recognition feature values for all candidate target regions from 174 windows.

a. Tanks.

54T	5	22	4.06	.55	.33	1.35	.41	.53	2.60	.52	2.43	3.71	.67	4.04
55T	1	27	3.69	.68	.46	1.33	.46	.02	2.98	.61	1.87	1.13	.67	3.36
55T	2	26	4.04	.38	.28	1.35	.39	.49	1.45	.40	1.19	1.42	.26	3.81
55T	5	23	3.88	.21	.28	1.26	.38	.23	1.95	.22	1.63	2.08	.38	3.20
57T	1	24	5.14	.89	.06	1.40	.39	.15	2.85	.19	1.61	4.08	.76	3.63
57T	5	22	4.40	.73	.06	1.64	.32	.28	4.83	.17	1.54	1.52	.33	1.07
58T	6	21	3.70	.72	.10	1.38	.29	.32	1.32	.32	1.85	3.56	.67	2.77
58T	9	21	3.18	.74	.02	1.30	.20	.22	2.77	.22	1.03	1.66	.63	2.37
61T	1	49	6.66	.21	.11	1.28	.38	.30	2.39	.08	.71	1.07	.58	2.62
61T	2	48	6.67	.62	.02	1.08	.38	.28	2.39	.21	.91	1.30	.60	2.62
61T	4	46	6.87	1.15	.25	1.28	.44	-.17	2.91	.15	1.07	2.30	.58	3.19
61T	5	45	6.95	1.28	.29	1.28	.45	-.12	1.75	.18	1.87	2.30	.58	3.09
61T	6	43	7.06	1.04	.23	1.05	.36	-.06	3.67	.19	2.46	4.09	.39	2.10
62T	2	26	3.76	.65	.06	1.72	.45	.16	2.37	.13	1.83	1.99	.76	6.35
62T	3	25	3.95	.93	.15	.93	.33	.37	4.02	.29	2.06	2.59	.78	4.44
62T	4	24	4.23	.73	.06	1.29	.41	.31	2.30	.29	2.30	3.49	.73	3.67
63T	1	55	6.03	.91	.12	1.51	.43	-.09	4.09	.32	2.21	3.45	.63	3.72
63T	2	54	6.10	1.03	.22	1.67	.43	-.03	6.18	.25	2.35	3.45	.56	3.72
63T	4	52	6.19	1.03	.25	1.25	.38	-.11	6.07	.05	2.75	3.59	.81	10.32
64T	4	31	4.06	1.38	.49	2.37	.38	-.31	4.73	.05	2.98	3.38	.80	9.66
65T	1	29	3.95	.50	.02	1.02	.37	.72	3.10	.71	1.31	1.99	.80	4.40
68T	3	27	4.30	.55	.14	1.02	.40	.53	3.94	.49	1.31	2.90	.73	3.61
68T	5	26	4.57	.48	.51	1.38	.44	.92	1.38	.92	2.09	3.25	.59	3.09
68T	1	29	3.87	.38	.67	1.36	.41	.74	1.60	.74	1.13	3.04	.65	3.18
69T	3	27	4.30	.51	.41	1.74	.42	.59	1.38	.59	1.57	2.34	.95	4.51
69T	3	26	4.52	.51	.40	1.10	.42	.59	.24	.59	1.92	2.80	.95	3.73
73T	5	43	6.29	.64	.03	1.29	.35	.13	3.20	.19	1.01	1.74	.60	3.98
73T	5	47	6.67	.66	.06	1.24	.33	.15	.24	.15	1.01	1.76	.60	2.98
73T	5	45	6.88	1.16	.33	1.12	.37	-.15	4.33	.22	1.70	3.49	.59	2.63
73T	6	44	6.96	1.13	.30	1.74	.46	-.11	2.71	.03	2.03	4.06	.49	2.50
73T	7	43	7.09	.42	.52	1.30	.34	.19	1.34	.10	2.52	4.75	.31	1.38
74T	2	48	6.73	.81	.01	.88	.32	.21	.63	.55	.65	.97	.46	3.09
74T	3	47	6.82	1.18	.35	1.26	.40	-.17	4.07	.16	.89	1.69	.70	3.26
74T	5	45	6.99	1.00	.20	1.63	.46	.00	.82	.01	1.55	3.27	.62	3.93
75T	3	26	3.89	.39	.58	1.63	.39	.82	1.08	.86	1.13	.85	.28	2.35
75T	4	28	4.01	.37	.07	1.03	.36	.48	1.55	.26	1.01	.85	.24	3.86
75T	5	27	6.36	.74	.07	1.00	.36	.29	1.09	.29	2.70	1.76	.58	3.29
75T	5	25	6.62	.61	.01	1.78	.36	.21	1.79	.30	2.19	2.64	.59	3.26
75T	7	24	6.70	1.01	.21	1.19	.37	-.01	4.29	.26	2.38	3.41	.55	3.64
75T	8	23	6.79	1.75	.55	1.61	.37	-.28	4.01	.18	2.67	4.30	.48	2.81
75T	10	23	6.87	1.24	.60	1.62	.39	-.21	3.81	.06	3.01	5.18	.30	3.09
76T	2	33	5.72	.33	.81	1.37	.33	.80	4.97	.72	1.04	.66	.35	2.37
76T	3	32	5.93	.39	.66	1.35	.32	.74	4.50	.68	1.28	1.62	.34	3.62
76T	4	31	6.18	.54	.11	2.40	.36	.00	3.08	.16	1.52	1.90	.59	3.62
76T	6	30	6.45	.97	.17	1.34	.35	.03	1.91	.31	1.91	2.80	.60	4.48
76T	7	28	6.53	.65	.19	1.21	.31	.47	2.56	.29	2.35	3.41	.58	4.40
76T	8	27	6.64	.70	.11	1.11	.34	.34	5.07	.16	2.69	3.84	.55	4.19
76T	9	26	6.71	.76	.04	1.02	.36	.26	5.68	.37	2.97	4.73	.54	3.96
76T	10	25	6.79	.90	.10	1.05	.36	.11	5.46	.61	3.28	5.55	.48	3.36
76T	11	24	6.84	.97	.17	1.76	.47	.03	5.63	.02	3.29	6.01	.41	2.79
76T	13	33	6.93	1.69	.38	1.20	.40	-.18	3.32	.26	3.86	6.94	.30	1.87
78T	3	32	5.97	.39	.67	1.50	.18	.74	2.46	.74	2.22	1.60	.27	1.56
78T	5	31	6.25	.81	.01	2.22	.18	.21	.54	.21	2.42	1.97	.30	1.87
78T	6	30	6.36	.62	.23	1.57	.18	.45	3.01	.43	2.62	1.89	.55	2.32
78T	7	29	6.40	.49	.24	1.22	.18	.61	3.40	.50	2.91	2.66	.38	2.63
78T	8	28	6.52	.45	.08	1.60	.20	.13	2.86	.06	3.16	3.34	.40	2.76
79T	1	30	5.75	.98	.18	1.83	.32	.02	2.81	.01	1.72	1.63	.44	2.04
79T	2	28	6.33	1.07	.50	1.07	.30	-.07	1.33	.09	1.80	2.13	.28	1.92
79T	5	29	6.47	.99	.14	1.58	.32	.01	1.23	.14	2.32	3.05	.23	2.25
79T	9	27	6.11	.22	.59	1.57	.25	.70	3.62	.67	1.56	1.70	.22	2.28
79T	8	25	6.85	.69	.17	1.21	.35	.39	3.03	.31	2.76	3.43	.50	2.20
79T	10	24	7.00	.59	.28	1.27	.35	.64	3.80	.60	3.03	2.35	.32	1.20
80T	1	32	3.21	1.16	.36	1.60	.30	-.15	6.98	.32	.65	.65	.36	1.36
80T	4	31	5.62	1.53	.37	1.08	.29	-.36	5.73	.56	.96	.90	.50	3.17
80T	6	30	6.06	1.06	.26	1.51	.27	-.07	4.44	.32	1.26	1.90	.49	2.65
80T	7	30	6.25	1.08	.77	1.78	.27	-.08	4.47	.07	1.59	2.08	.50	2.83
80T	8	28	6.50	1.05	.24	2.11	.40	-.05	3.16	.13	1.84	2.71	.51	2.99
80T	9	27	6.54	1.41	.52	1.48	.34	-.33	2.85	.31	2.19	3.19	.51	2.97
80T	10	26	6.85	1.01	.21	1.50	.36	-.01	3.56	.01	2.68	3.72	.46	2.08
80T	11	26	3.97	.22	.59	1.72	.26	.70	3.65	.64	1.17	1.56	.41	2.63
80T	13	24	6.92	.75	.03	1.53	.35	.25	3.09	.21	3.09	4.79	.42	3.44
80T	14	23	5.90	.64	.05	1.55	.36	.17	3.88	.11	3.34	5.31	.43	1.33
80T	15	22	7.10	.96	.17	1.79	.39	.04	4.04	.23	3.73	5.78	.30	1.63
89T	1	32	5.82	.04	.04	1.89	.16	.18	1.86	.17	2.11	1.99	.30	1.42
89T	2	31	6.01	.72	.04	1.01	.21	.32	1.06	.32	2.33	1.76	.28	3.79
89T	3	30	6.20	.76	.03	1.15	.26	.26	4.95	.20	2.67	2.42	.51	3.18
89T	4	28	6.32	.99	.15	1.57	.27	.01	1.66	.02	2.63	3.37	.41	2.35
89T	5	28	6.53	.91	.14	1.66	.35	.09	1.50	.12	2.91	3.38	.50	3.11
89T	7	26	6.71	1.18	.15	1.31	.42	-.18	3.91	.11	3.24	4.30	.55	3.30
89T	8	25	6.76	1.06	.26	1.38	.38	-.06	3.20	.13	3.51	5.23	.42	2.40
89T	9	24	6.80	.83	.03	1.24	.35	.17	2.71	.05	3.79	5.93	.38	1.63
89T	10	23	6.81	.63	.39	1.05	.30	.37	3.15	.35	4.24	6.66	.30	1.52
92T	2	32	5.70	.53	.17	1.66	.28	.56	4.31	.49	.94	1.10	.35	1.50
92T	3	31	5.95	.64	.19	1.54	.35	.41	2.47	.47	1.19	1.67	.42	2.19
92T	4	30	6.16	.76	.07	1.29	.32	.29	4.37	.11	1.69	2.36	.61	3.74
92T	5	29	6.32	.88	.10	1.61	.43	.12	1.64	.35	1.80	3.18	.54	3.90
92T	6	27	6.40	.95	.09	1.48	.43	.10	.67	.13	2.25	3.38	.67	4.85
92T	8	25	6.51	.91	.24	1.35	.45	.09	5.53	.08	2.48	4.20	.57	4.02
92T	9	25	6.57	1.05	.22	1.35	.43	-.05	5.91	.15	2.78	4.95	.54	3.36
92T	10	24	6.64	1.04	.04	1.74	.49	-.17	5.62	.15	3.16	5.65	.46	3.40
92T	11	23	6.72	.59	.19	1.25	.41	.41	2.60	.40	3.60	6.55	.36	2.05
92T	12	22	6.86	.74	.04	1.17	.37	.26	4.03	.19	4.16	7.16	.27	2.05
95T	1	33	5.97	.54	.35	1.19	.22	.54	4.33	.49	2.25	2.14	.38	3.06
95T	2	32	6.14	.92	.61	1.62	.32	.08	1.90	.63	2.37	2.13	.45	2.66
95T	3	31	6.25	.92	.26	1.66	.35	.14	1.84	.35	2.52	2.06	.65	3.11
95T	4	30	6.44	1.20	.37	1.36	.39	-.18	1.86	.35	2.66	2.87	.56	4.01
95T	5	29	6.57	1.06	.25	1.61	.43	-.06	1.92	.04	2.90	3.63	.56	4.01
95T	6	28	6.63	1.07	.26	1.56	.45	-.07	2.60	.05	3.08	4.02	.53	3.58
95T	8	26	6.78	.92	.12	1.09	.37	.09	5.61	.44	3.55	5.70	.53	3.40
95T	9	25	6.87	.95	.16	1.03	.47	.05	1.06	.05	3.96	6.46	.45	2.71
95T	10	24	6.94	.65	.18	1.22	.40	.37	3.50	.35	4.23	6.64	.35	2.02
99T	1	32	5.84	.67	.16	2.12	.36	.38	2.22	.34	1.25	.68	.38	1.90
99T	2	31	6.01	.73	.09	1.28	.31	.28	1.67	.28	1.42	1.26	.47	3.19
99T	3	30	6.23	.93	.15	1.25	.39	.09	1.28	.11	1.75	1.91	.62	3.80
99T	4	29	6.36	.85	.16	1.79	.45	.20	1.71	.19	1.95	2.62	.62	3.60
99T	5	27	6.49	.71	.08	1.69	.50	.31	1.70	.22	2.30	3.36	.67	4.80
99T	6	26	6.55	1.20	.61	1.69	.44	-.22	1.63	.22	2.60	4.37	.59	4.17
99T	8	25	6.61	.59	.19	.92	.36	.61	5.97	.58	2.93	5.35	.54	3.83
99T	9	24	6.67	.61	.01	.86	.33	.21	6.34	.53	3.28	6.20	.44	3.10
99T	10	23	6.74	.74	.06	.92	.33	.28	4.32	.22	3.62	6.76	.34	2.40
99T	11	22	6.90	.99	.53	1.83	.43	.03	2.44	.15	4.01	6.36	.30	2.08
105T	1	34	5.22	.20	.06	1.61	.30	.05	2.51	.65	2.31	1.36	.30	1.89
105T	4	33	6.01	.97	.04	1.08	.29	.05	1.31	.07	2.02	1.97	.37	1.69
105T	5	32	6.23	.97	.17	1.88	.35	.03	2.21	.04	2.06	2.33	.36	2.39
105T	6	31	6.37	.55	.33	1.95	.24	.55	3.61	.44	2.21	2.49	.59	2.60
105T	7	30	6.47	.89	.09	1.40	.37	.12	3.01	.03	2.40	2.89	.46	3.33
105T	8	29	6.56	.82	.02	1.81	.35	.20	6.86	.04	2.65	3.49	.58	4.16
105T	9	28	6.63	.88	.08	1.03	.33	.13	3.68	.40	2.88	4.66	.52	3.04
105T	10	28	6.83	.87	.07	1.05	.33	.13	2.01	.06	3.19	4.96	.57	3.65
105T	12	25	6.88	1.09	.16	1.39	.41	-.27	5.51	.12	2.95	6.32	.57	3.03
105T	13	34	6.95	1.49	.16	1.80	.45	-.36	5.31	.14	2.65	2.45	.57	3.52
109T	1	34	3.33	1.64	.46	2.18	.46	-.46	5.70	.02	2.05	2.15	.58	6.09
109T	2	34	3.00	1.39	.50	1.53	.40	-.31	3.69	.25	3.10	3.17	.56	6.67
109T	3	33	3.81	2.08	.89	2.65	.42	-.63	4.08	.58	3.01	3.82	.61	6.50
109T	17	30	6.78	.84	.36	4.32	.37	.02	3.04	.32	3.82	3.96	.51	3.20
110T	1	29	6.10	.71	.29	1.28	.33	.07	2.06	.24	3.32	2.98	.28	3.38
110T	2	29	6.39	1.01	.21	1.31	.29	-.01	4.06	.25	2.16	3.08	.60	3.04
110T	4	27	6.60	.69	.13	1.31	.41	.35	3.00	.33	2.51	3.35	.55	2.62
110T	5	26	6.69	1.01	.21	1.14	.37	-.01	5.15	.37	2.74	4.15	.47	2.38
110T	6	25	6.77	.92	.12	1.08	.38	.08	5.45	.61	2.99	4.85	.39	2.01
110T	7	24	6.89	1.13	.31	1.25	.40	-.12	3.49	.12	3.38	5.53	.26	1.34
114T	1	33	5.72	.67	.07	1.91	.32	.14	.87	.14	1.12	1.19	.27	1.04
114T	2	32	6.01	.63	.07	1.66	.37	.31	2.32	.14	1.36	1.27	.42	2.12
114T	4	31	6.23	.49	.44	1.95	.33	.61	2.23	.10	1.80	1.98	.62	2.36
114T	5	30	6.43	.62	.02	1.59	.43	.19	.69	.19	2.23	3.00	.51	3.37
114T	6	28	6.54	.64	.20	1.05	.36	.42	4.61	.28	2.59	4.41	.60	3.02
114T	8	27	6.70	1.31	.21	1.23	.42	-.01	4.99	.35	2.97	5.06	.60	2.18
114T	9	26	6.81	.00	.27	.76	.31	.48	6.58	.67	3.35	5.79	.28	1.22
114T	10	25	6.91	2.26	.97	.93	.35	-.67	4.08	.61	.08	.60	.57	1.22
122T	1	26	4.10	1.02	.22	1.72	.43	-.02	2.38	.08	1.61	1.52	.62	2.22
122T	3	25	4.99	.39	.03	1.76	.47	.21	2.38	.19	1.70	2.42	.44	1.62
122T	4	22	5.32	.15	.02	1.68	.39	.34	3.56	.18	.91	.95	.57	3.22
123T	3	29	2.92	.82	.06	1.38	.28	.16	5.00	.18	.91	.95	.57	3.22
123T	5	26	4.71	.60	.00	1.39	.47	-.10	3.06	.21	1.19	1.92	.78	4.87
123T	5	22	5.03	1.11	.28	1.32	.41	-.10	3.67	.16	1.19	1.92	.78	4.49
123T	8	23	5.27	1.28	.61	1.07	.35	-.22	6.30	.15	2.20	3.31	.57	3.63
123T	9	22	5.43	1.22	.36	1.37	.42	-.20	2.09	.19	2.35	3.67	.63	3.42
124T	1	28	4.29	.67	.16	2.10	.44	.38	2.58	.37	1.08	.80	.50	3.60

Table 3.9.4 (continued)

1241	2	27	4.47	.53	.37	1.64	.43	.56	1.39	.56	1.27	1.09	.61	4.35
1241	4	25	4.81	.68	.13	1.90	.46	.37	2.87	.35	1.64	4.34	.76	5.03
1241	5	24	4.96	.54	.19	1.69	.48	.41	2.01	.41	1.17	3.00	.71	4.85
1241	6	23	5.11	.67	.08	1.75	.46	.14	1.38	.17	2.48	3.58	.63	4.29
1241	7	22	5.16	1.05	.22	1.48	.44	-.03	2.51	.09	2.58	4.11	.57	2.56
1251	2	27	4.62	.53	.21	1.86	.41	.43	3.69	.38	1.10	1.22	.48	2.09
1251	3	26	4.69	.34	.04	1.92	.48	.18	4.06	.15	1.42	1.70	.75	3.33
1251	5	24	5.36	.68	.09	1.31	.43	.12	2.54	.08	1.94	4.70	.61	2.81
1261	1	27	4.65	.74	.07	1.89	.45	.30	2.48	.28	1.08	1.51	.33	1.45
1261	2	26	5.04	1.71	.21	.85	.33	-.01	6.23	.55	1.45	1.96	.66	3.00
1261	4	24	5.47	.71	.10	1.55	.41	.32	1.92	.32	1.91	2.91	.62	2.98
1261	5	23	5.71	.81	.01	1.45	.40	.21	3.31	.21	2.24	3.17	.47	2.07
1271	2	30	4.53	.60	.26	1.99	.42	.47	2.08	.46	1.06	.51	.31	2.53
1271	3	29	4.80	.52	.38	1.58	.39	.57	.85	.57	1.32	1.33	.52	3.80
1271	4	28	4.93	.76	.05	1.81	.42	.27	2.85	.25	1.52	2.02	.61	4.31
1271	5	27	5.10	.46	.51	1.06	.33	.05	3.33	.64	1.86	4.61	.62	4.22
1271	7	25	5.60	.86	.07	1.41	.43	.15	5.17	.13	2.51	4.88	.87	6.54
1271	8	24	5.54	1.03	.22	.93	.31	-.03	5.29	.19	2.83	4.18	.53	3.59
1281	2	28	4.25	.64	.20	1.61	.40	.41	1.92	.41	1.13	1.45	.41	2.75
1281	3	27	4.57	.74	.07	1.42	.37	.30	4.51	.07	1.56	2.03	.60	3.60
1281	5	25	5.11	.89	.10	1.35	.35	.13	6.81	.11	2.09	4.92	.57	3.83
1281	6	24	5.33	.83	.03	1.52	.41	.19	5.58	.05	2.34	5.04	.53	2.91
1291	7	23	5.65	.93	.14	1.65	.34	.07	6.54	.28	2.63	5.42	.47	1.82
1291	2	30	4.72	.72	.10	1.56	.37	.32	3.93	.16	.89	1.04	.40	3.03
1291	4	28	4.97	.86	.07	1.79	.42	.15	.45	.15	1.98	1.97	.69	2.99
1291	6	28	5.11	.80	.01	1.66	.47	.22	.35	.22	1.42	1.79	.90	0.45
1291	6	26	5.36	.64	.19	1.19	.40	.41	4.15	.39	2.03	3.17	.80	5.63
1291	7	25	5.47	.94	.15	1.23	.43	.06	5.11	.12	2.36	3.98	.65	4.55
1291	8	24	5.57	.99	.19	1.77	.42	.01	5.38	.16	2.69	4.61	.47	3.38
1301	2	29	4.81	.58	.24	1.98	.44	.50	3.80	.55	1.10	1.11	.56	3.83
1301	3	28	4.81	.49	.25	1.55	.45	.62	3.64	.59	1.34	1.78	.65	4.42
1301	4	27	4.96	.75	.06	1.94	.42	.28	2.00	.28	1.60	2.49	.70	4.88
1301	6	25	5.19	.70	.12	.82	.51	.34	5.20	.65	2.15	3.35	.67	4.52
1311	1	30	3.37	1.50	.01	1.20	.44	-.42	2.09	.51	.81	1.58	.54	1.44
1311	2	29	0.02	.62	.22	1.05	.18	.44	5.81	.19	.87	.46	.39	.97
1311	4	27	0.34	.50	.43	1.08	.28	.61	5.58	.58	1.34	1.65	.41	1.07
561	2	22	3.47	.95	.15	1.63	.40	.05	4.22	.25	.93	1.91	.58	1.92

Table 3.9.4 (continued)

TRUCKS

WINDOW	CC#	THRESH	LOG A	H/W	(H/W)"	H•W/A	(H•W)/P	DIFF	SKEW	ASYM	SDEV	GL DIFF	E&P	EP
4R	7	23	4.93	.46	.50	1.95	.45	.65	2.14	.65	1.76	2.29	.68	3.29
6R	4	26	4.78	.51	.41	1.14	.35	.59	5.17	.46	2.01	1.92	.51	2.71
9R	5	25	4.89	.40	.63	1.55	.36	.72	1.43	.72	1.19	.85	.73	2.37
18R	2	28	6.05	.92	.12	1.14	.33	.09	.83	.09	.84	1.15	.56	1.09
22R	4	26	6.00	.40	.00	2.30	.35	.22	4.55	.20	1.49	1.54	.36	.96
24R	6	30	6.12	1.09	.28	1.23	.33	-.08	2.06	.06	1.72	2.03	.53	2.10
26R	2	29	4.58	.74	.07	1.31	.36	.30	3.52	.24	.70	.55	.59	1.49
31R	3	28	4.69	.86	.06	1.20	.40	.16	5.11	.33	.96	1.38	.79	2.32
41R	6	30	6.15	1.12	.30	1.88	.41	-.11	4.08	.21	1.81	1.89	.57	2.80
47R	11	25	6.39	1.53	.59	.86	.38	-.40	3.90	.34	3.44	5.85	.54	4.23
57R	8	24	4.77	.71	.11	1.60	.41	.33	5.01	.16	1.55	1.80	.39	1.46
71R	4	45	6.79	.57	.30	1.45	.41	.51	2.75	.49	1.27	2.35	.78	3.54
72R	2	47	6.71	.66	.18	2.05	.53	.40	4.07	.32	.66	1.28	.84	3.16
77R	7	28	6.61	.29	.95	.90	.36	.64	1.70	.84	2.03	3.04	.63	3.49
100R	4	26	5.67	1.12	.30	1.16	.37	-.11	4.57	.27	1.81	2.54	.62	2.77
104R	4	30	5.62	.75	.06	1.64	.42	.28	2.26	.27	1.94	2.49	.65	4.01
109R	5	29	5.50	.38	.68	1.29	.24	.75	2.95	.74	1.94	1.79	.50	2.23
132R	4	28	6.49	.38	.68	1.16	.15	.75	4.43	.70	1.52	.90	.37	1.22
133R	4	27	6.38	1.01	.21	1.18	.20	-.01	6.08	.52	1.82	1.53	.46	1.23
134R	5	27	6.62	.76	.04	2.07	.25	.26	6.27	.49	1.46	1.16	.54	1.27
135R	3	27	6.24	1.07	.26	1.59	.39	-.07	2.83	.09	1.16	1.34	.63	2.00
3R	1	31	3.00	1.22	.35	1.55	.42	-.20	4.07	.12	1.09	.39	.77	3.88
33R	1	27	3.26	.72	.10	1.73	.44	.32	4.79	.04	1.05	1.05	.77	2.23
34R	2	27	3.91	.88	.08	1.47	.40	.13	3.35	.09	1.20	1.83	.76	2.58
53R	1	25	3.53	.76	.05	1.04	.35	.27	5.03	.23	.93	1.25	.65	1.83
54R	1	24	3.66	.99	.19	1.38	.38	.01	3.09	.13	.84	.99	.82	2.00
55R	2	25	3.61	1.12	.30	1.61	.48	-.11	1.78	.10	1.17	1.93	.79	2.54
58R	4	25	3.89	.73	.08	1.45	.47	.30	1.49	.30	1.52	2.20	.71	3.61
59R	2	25	3.50	.68	.15	1.62	.52	.37	3.75	.31	1.03	1.55	.76	2.76
3R	12	25	6.64	.61	.25	2.23	.34	.46	3.07	.44	2.75	1.32	.35	1.22
3R	13	25	5.11	.79	.01	1.63	.30	.23	3.77	.10	1.06	.88	.61	1.78
3R	14	24	6.83	.42	.58	1.60	.30	.69	1.02	.69	2.91	1.24	.42	1.36
3R	15	24	5.39	1.04	.23	2.15	.33	-.04	2.84	.11	1.25	1.24	.60	1.66
3R	16	23	7.28	.98	.19	1.96	.22	.02	1.20	.01	2.95	1.93	.42	1.03
4R	3	26	4.14	.38	.68	1.66	.41	.74	3.73	.72	1.10	1.25	.52	2.87
4R	4	25	4.49	.47	.48	1.82	.43	.64	3.03	.62	1.28	1.69	.65	3.56
4R	5	24	4.69	.35	.76	1.37	.39	.78	4.15	.74	1.43	1.87	.59	3.15
4R	9	22	5.20	.44	.54	1.77	.44	.67	2.23	.67	2.10	2.85	.60	2.65
6R	1	29	3.61	.89	.10	1.15	.32	.12	2.54	.07	1.37	1.78	.46	2.96
6R	2	28	3.91	.75	.06	1.53	.44	.28	2.51	.26	1.59	1.94	.45	2.87
6R	3	27	4.29	.49	.44	1.57	.37	.61	3.35	.59	1.85	2.19	.48	2.87
6R	5	25	5.02	.59	.27	1.70	.44	.48	2.61	.47	2.12	2.26	.41	2.15
6R	7	24	5.17	.47	.47	1.05	.39	.63	5.46	.47	2.26	2.70	.42	2.11
6R	9	23	5.89	.30	.90	1.39	.28	.83	5.51	.71	2.39	2.78	.38	1.25
9R	1	27	3.33	.80	.00	1.52	.36	.22	4.48	.19	.78	.21	.46	1.62
9R	2	26	4.41	.46	.50	1.63	.33	.65	1.60	.65	1.01	1.09	.56	1.73
9R	11	24	5.19	.62	.23	1.67	.33	.45	2.79	.44	1.46	1.75	.56	1.85
18R	3	27	6.34	1.16	.34	1.15	.37	-.15	.89	.15	1.12	1.70	.46	.82
22R	1	29	4.39	.60	.25	2.05	.43	.47	2.65	.45	.68	.67	.44	1.19
22R	2	28	4.88	.41	.61	1.48	.37	.71	1.67	.71	.87	1.08	.72	1.90
22R	3	27	5.69	.92	.13	1.87	.29	.08	3.65	.17	1.32	1.01	.44	1.25
22R	5	25	6.46	1.29	.43	1.28	.24	-.25	3.13	.21	1.71	1.68	.30	.84
24R	2	33	4.84	.59	.27	1.45	.32	.48	1.81	.45	1.32	1.96	.44	1.64
24R	4	32	5.64	1.03	.23	1.86	.30	-.03	3.55	.18	1.35	1.44	.41	1.37
24R	5	31	5.91	.99	.19	1.35	.27	.01	1.76	.04	1.50	1.60	.39	1.50
24R	8	29	6.40	.87	.06	1.88	.36	.16	4.12	.20	1.99	2.79	.44	1.59
24R	9	28	6.68	.74	.07	1.81	.33	.29	3.78	.21	2.26	3.03	.43	1.37
31R	2	29	4.33	.73	.09	1.16	.38	.31	5.03	.18	.64	.62	.72	2.26
32R	3	23	3.74	.62	.23	1.78	.39	.45	1.57	.45	.86	1.23	.71	2.00
32R	4	22	4.37	1.00	.20	2.16	.39	.00	3.76	.20	1.05	1.07	.74	1.93
33R	2	26	4.03	.56	.33	1.35	.43	.53	3.80	.49	1.28	1.88	.64	1.62
34R	1	28	3.43	.80	.00	1.52	.45	.22	2.37	.21	.86	1.21	.91	3.14
34R	3	26	4.22	1.01	.21	1.09	.33	-.01	5.75	.47	1.43	2.34	.52	1.65
35R	1	25	3.87	1.39	.50	1.36	.37	-.32	2.96	.29	.60	.88	.58	1.00
41R	4	32	5.70	1.43	.53	1.83	.21	-.34	2.89	.32	1.33	1.25	.31	1.27
41R	5	31	5.96	1.97	.84	1.26	.23	-.59	.77	.59	1.58	1.64	.42	1.79
41R	7	29	6.31	.87	.07	1.23	.36	.14	2.94	.07	2.06	2.67	.54	2.75
47R	3	33	5.44	1.52	.59	1.47	.36	-.40	3.56	.35	1.30	1.73	.28	2.20
47R	4	32	5.66	1.29	.43	1.56	.43	-.25	3.59	.17	1.49	1.85	.33	2.84
47R	5	31	5.79	1.24	.39	1.32	.41	-.21	2.55	.19	1.68	2.31	.28	2.65
47R	6	30	5.94	1.27	.42	1.57	.45	-.24	3.69	.14	1.99	2.92	.29	2.88
47R	7	29	6.05	1.15	.32	1.25	.41	-.14	1.60	.13	2.26	3.42	.38	3.50
47R	8	28	6.15	1.35	.47	1.19	.45	-.29	1.54	.29	2.54	3.82	.47	4.18
47R	9	27	6.22	1.12	.31	1.43	.48	-.12	2.88	.01	2.77	4.25	.52	4.37
47R	10	26	6.30	2.05	.87	1.00	.44	-.62	2.18	.61	3.10	5.15	.53	4.38
51R	1	27	3.04	.47	.65	.82	.31	.63	1.50	.61	1.08	1.65	.69	2.19
51R	2	26	3.71	.68	.09	1.51	.45	.12	4.15	.21	1.32	2.16	.69	2.42
52R	1	25	3.33	.75	.06	1.47	.43	.28	3.21	.24	1.13	1.33	.67	2.29
52R	2	24	3.56	.50	.43	1.18	.36	.60	3.34	.58	1.38	2.11	.58	2.08
53R	2	24	4.78	.40	.64	1.73	.40	.73	2.36	.72	.98	1.02	.57	1.09
55R	1	26	3.14	1.61	.64	1.23	.41	-.45	1.35	.44	.96	1.59	.77	2.53
55R	3	24	4.06	1.66	.67	1.14	.39	-.47	1.76	.45	1.39	2.18	.48	1.67
56R	1	29	3.16	.91	.14	1.01	.33	.07	2.36	.04	1.10	1.61	.56	2.78
56R	2	28	3.47	.66	.17	1.11	.35	.39	.94	.39	1.27	1.51	.61	3.09
56R	3	27	3.83	.48	.47	.54	.24	.63	6.51	.18	1.56	2.02	.57	3.00
56R	4	27	3.26	1.22	.36	1.16	.35	-.20	4.17	.14	.57	.70	.55	1.70
56R	5	26	4.13	.25	1.00	.50	.29	.66	3.87	.63	1.79	2.69	.48	2.61
56R	6	26	3.07	1.35	.47	1.32	.40	-.29	4.80	.15	.83	1.10	.81	2.36
56R	7	25	4.84	.90	.10	2.34	.39	.11	6.21	.53	1.88	2.76	.37	1.49
56R	9	24	5.43	.35	.77	1.75	.32	.78	6.53	.50	1.85	1.97	.43	1.26
57R	3	27	5.30	.63	.03	1.20	.35	.19	1.41	.19	.74	.79	.62	2.57
57R	5	26	3.71	.70	.12	1.43	.41	.34	1.90	.34	1.08	1.56	.68	2.64
57R	6	25	4.20	.68	.08	1.44	.35	.13	4.36	.24	1.33	2.08	.64	2.49
58R	1	27	3.37	.78	.02	1.30	.38	.24	3.25	.19	.91	1.42	.82	3.96
58R	2	26	3.61	.81	.02	1.55	.49	.20	2.59	.18	1.17	1.53	.78	3.83
58R	9	24	4.20	.56	.33	1.67	.46	.53	2.54	.52	1.84	2.91	.49	2.29
59R	1	26	3.09	.40	.64	.95	.34	.73	2.22	.72	.73	.89	.50	1.61
59R	4	24	3.89	.71	.11	1.15	.40	.33	1.23	.33	1.33	2.01	.54	2.18
59R	5	23	5.14	.30	.92	2.44	.39	.84	3.04	.63	1.28	1.77	.47	1.28
71R	2	47	6.57	.31	.86	1.00	.34	.82	1.92	.62	.72	.88	.47	2.24
71R	3	46	6.67	.57	.30	1.60	.48	.51	3.80	.47	.93	1.51	.70	3.33
71R	5	44	6.88	.50	.42	1.11	.35	.59	1.32	.59	1.59	2.96	.68	3.18
72R	4	46	6.53	.44	.54	1.20	.43	.67	2.36	.67	1.03	2.19	.81	3.16
77R	3	32	5.99	.48	.45	2.38	.31	.62	3.12	.60	1.01	.99	.25	1.37
77R	4	31	6.18	.49	.44	2.00	.36	.61	3.00	.60	1.21	1.23	.45	2.48
77R	5	30	6.37	.66	.17	1.82	.37	.39	2.71	.38	1.48	1.92	.61	3.19
77R	6	29	6.51	.60	.25	1.96	.41	.47	3.68	.42	1.77	2.53	.62	3.42
77R	8	27	6.71	.38	.68	.76	.32	.75	3.65	.72	2.34	3.59	.52	2.93
100R	1	26	4.90	1.03	.23	1.45	.32	-.03	.90	.03	1.22	1.56	.27	1.05
100R	2	28	5.25	.69	.13	1.27	.33	.35	2.51	.36	1.39	1.54	.37	1.73
100R	3	27	5.54	.81	.02	1.45	.38	.23	2.56	.18	1.64	2.21	.59	2.59
100R	5	25	5.85	1.00	.20	1.07	.39	.00	4.53	.29	2.11	3.01	.52	2.43
100R	6	24	6.00	1.28	.42	1.10	.40	-.24	4.19	.07	2.41	3.54	.32	1.39
104R	1	33	5.09	.75	.06	1.65	.26	.28	5.74	.37	1.36	1.51	.44	2.27
104R	2	32	5.35	.90	.11	2.13	.36	.10	1.67	.09	1.52	1.49	.53	3.15
104R	3	31	5.48	.58	.29	1.20	.28	.50	5.16	.32	1.69	2.23	.61	3.61
104R	5	29	5.76	1.03	.23	1.23	.39	-.03	4.67	.31	2.25	2.73	.63	4.00
104R	6	28	5.88	.84	.05	1.66	.43	.17	1.35	.17	2.54	3.52	.62	3.96
104R	7	27	6.03	.70	.12	1.23	.38	.34	2.19	.34	2.91	4.42	.43	2.85
109R	1	32	4.09	2.07	.88	2.11	.38	-.62	2.63	.61	1.41	.52	.58	2.29
109R	2	31	4.62	.85	.06	2.30	.31	.16	4.37	.22	1.50	1.44	.47	1.79
109R	3	30	4.82	.97	.18	2.46	.35	.03	3.03	.13	1.68	1.14	.45	1.85
109R	4	30	3.74	1.48	.56	2.23	.46	-.38	1.95	.37	1.64	.86	.59	2.24
109R	6	28	5.75	.42	.59	1.09	.26	.70	2.75	.70	2.10	2.05	.49	2.31
109R	7	27	6.03	.80	.00	1.97	.31	.22	2.44	.21	2.27	2.64	.42	1.93
132R	3	29	6.37	.53	.37	1.52	.15	.56	5.11	.42	1.43	.56	.34	1.12
132R	6	27	6.85	.69	.13	2.33	.23	.35	2.32	.35	1.83	1.62	.40	1.25
133R	2	29	5.71	.80	.00	2.09	.20	.22	4.36	.15	1.38	.50	.39	1.00
133R	3	28	6.01	1.06	.25	2.11	.18	-.05	2.57	.04	1.51	.83	.38	1.01
134R	1	30	4.77	2.09	.89	1.98	.31	-.63	1.02	.63	.74	.21	.37	1.02
134R	3	29	5.92	.65	.19	1.70	.22	.41	2.15	.40	.91	.65	.37	1.09
134R	4	28	6.13	.71	.11	.98	.20	.34	5.86	.37	1.09	.87	.52	1.54
135R	1	29	5.72	.65	.18	1.47	.18	.40	4.04	.33	.70	.43	.32	.83
135R	2	28	6.03	.82	.02	1.15	.27	.20	5.31	.35	.90	.78	.50	1.59
135R	4	26	6.40	1.11	.29	1.41	.40	-.10	3.47	.14	1.45	2.17	.44	1.35

Table 3.9.4 (continued)

b. Trucks.

APC'S

WINDOW	CC#	THRESH	LOG A	H/W	$(H/W)^2$	H*W/A	(H+W)/P	DIFF	SKEW	ASYM	SDEV	GL DIFF	EBP	EP
21A	5	27	5.16	.65	.50	1.50	.41	.65	3.74	.62	1.80	2.15	.71	3.56
22A	5	33	5.18	.52	.39	1.60	.40	.58	4.84	.47	1.63	3.41	.56	3.05
24A	6	29	5.55	.49	.44	1.76	.44	.61	3.59	.58	1.65	2.28	.72	3.49
27A	2	25	4.62	.67	.15	1.70	.44	.38	3.77	.32	.79	1.11	.78	2.24
37A	5	30	5.75	1.29	.43	1.75	.43	-.25	3.90	.12	1.85	2.91	.66	3.26
38A	6	26	5.65	.92	.13	1.67	.32	.08	4.27	.24	1.46	1.92	.56	1.85
42A	6	26	5.50	1.35	.47	1.82	.40	-.29	3.90	.20	1.54	2.20	.55	2.47
44A	6	32	6.51	1.07	.26	1.46	.42	-.07	3.77	.19	1.63	2.38	.51	2.00
46A	5	27	5.65	1.21	.37	2.01	.43	-.19	4.62	.24	2.04	2.88	.65	2.87
48A	5	28	5.71	1.12	.31	1.48	.32	-.12	2.17	.10	1.55	2.22	.57	2.13
50A	2	24	4.56	.61	.25	1.03	.35	.46	4.46	.37	.93	1.00	.55	1.06
57A	6	24	4.62	1.10	.29	1.56	.42	-.09	2.00	.07	1.55	1.77	.62	2.50
61A	6	44	5.68	1.69	.69	1.67	.39	-.48	2.69	.47	1.55	2.20	.82	4.50
73A	3	46	5.58	1.39	.50	1.78	.35	-.32	2.55	.30	1.31	2.09	.78	3.79
74A	4	46	5.71	1.12	.30	1.62	.35	-.11	2.59	.06	1.37	2.10	.79	4.63
75A	6	26	5.68	1.13	.31	2.15	.42	-.12	4.30	.23	2.51	3.56	.73	4.53
76A	5	30	5.45	1.03	.23	2.13	.39	-.03	3.49	.17	2.00	2.79	.61	3.84
79A	9	26	5.80	1.26	.41	1.82	.36	-.23	4.15	.09	2.40	3.05	.50	2.24
80A	4	27	5.51	1.11	.30	1.97	.39	-.11	4.57	.28	2.00	3.03	.80	4.35
90A	6	29	5.84	1.08	.27	1.55	.30	-.08	2.64	.06	2.28	2.53	.63	3.97
91A	5	30	5.74	.86	.07	1.69	.36	.15	2.63	.11	2.34	3.14	.66	4.46
93A	7	29	6.02	.75	.06	1.76	.31	.28	1.55	.28	3.21	1.85	.59	4.47
94A	6	28	5.85	.87	.08	1.08	.35	.14	5.68	.44	2.55	3.45	.70	4.45
96A	2	31	5.78	.89	.10	1.98	.34	.11	3.30	.11	2.26	.82	.67	4.21
97A	6	26	6.12	.96	.16	1.66	.35	.04	2.32	.06	3.19	3.97	.60	4.19
98A	5	24	6.07	.95	.16	1.51	.69	.05	2.87	.11	3.67	3.33	.63	4.40
101A	5	44	6.05	.87	.07	1.39	.44	.14	1.93	.13	1.67	2.33	.62	4.73
102A	2	47	5.35	.66	.17	1.04	.34	.39	4.25	.29	.82	1.13	.65	2.44
111A	3	29	5.39	.90	.10	1.88	.40	.11	3.81	.17	2.10	2.73	.74	4.96
112A	5	25	5.52	1.56	.61	1.69	.29	-.42	2.00	.41	2.69	2.10	.63	4.86
113A	7	24	6.06	1.02	.21	1.33	.42	-.02	1.81	.04	3.00	3.80	.64	4.57
114A	7	29	6.17	1.09	.28	1.28	.21	-.08	4.13	.23	1.83	1.46	.66	2.77
115A	6	29	5.65	1.00	.20	1.91	.35	.00	3.10	.14	2.21	3.19	.63	4.34
123A	5	25	4.67	.23	.38	1.06	.35	.57	.69	.57	1.71	2.62	.92	4.13
28A	2	29	4.06	.76	.05	1.24	.39	.27	3.48	.21	.72	.89	.79	1.85
32A	2	25	4.09	.59	.28	1.58	.44	.49	1.33	.49	.96	1.15	.78	1.86
33A	2	25	4.16	1.51	.58	1.32	.37	-.39	2.00	.39	.50	.48	.75	1.22
35A	1	26	3.37	.92	.12	1.54	.39	.08	3.99	.21	1.21	2.29	.75	1.92
51A	2	27	3.30	.77	.04	1.21	.38	.26	2.70	.24	.92	1.37	.55	4.25
52A	1	26	3.00	1.11	.29	1.29	.39	-.10	2.36	.06	.66	1.12	.75	3.37
53A	5	24	4.19	.57	.30	1.50	.48	.51	3.66	.47	1.61	2.20	.63	2.40
54A	3	25	3.89	.57	.31	1.01	.38	.51	2.75	.50	1.36	1.80	.86	3.43
55A	1	26	3.26	.89	.10	1.46	.42	.12	1.60	.11	.86	1.32	.60	2.20
56A	3	29	4.22	.46	.51	1.05	.38	.66	.67	.66	1.38	2.02	.81	3.62
58A	3	27	3.97	.64	.20	1.21	.40	.42	1.29	.42	1.60	2.21	.87	4.84
59A	5	27	4.16	.86	.18	1.52	.48	.40	1.77	.39	1.57	2.16	.79	4.24
122A	5	24	4.26	1.02	.22	1.62	.48	-.02	3.71	.19	.80	.85	.68	2.00
125A	2	26	4.22	.83	.04	1.49	.41	.18	3.85	.10	1.22	1.78	.77	3.47
127A	2	26	4.32	.43	.57	1.19	.42	.69	3.42	.67	2.15	3.17	.87	5.67
129A	2	28	4.44	.56	.35	1.51	.44	.55	4.35	.49	1.34	1.82	.71	3.07
130A	2	24	4.25	.80	.00	1.64	.47	.22	.70	.22	1.27	1.26	.87	3.27
21A	3	29	4.65	.81	.01	1.55	.39	.21	4.67	.23	1.30	1.45	.46	2.80
21A	4	28	4.84	.99	.19	1.26	.37	.01	5.47	.42	1.48	1.69	.67	3.47
21A	6	26	5.42	.58	.28	1.76	.42	.49	3.23	.47	2.07	2.79	.69	3.24
22A	3	25	4.70	.69	.13	1.83	.44	.36	3.32	.32	1.23	2.65	.37	2.20
22A	4	24	5.03	.60	.26	1.50	.36	.47	5.31	.26	1.42	3.45	.47	2.67
22A	6	22	5.36	.62	.23	1.44	.36	.45	4.02	.39	1.95	4.08	.55	2.81
24A	1	32	4.78	.73	.08	1.62	.36	.30	5.50	.30	.80	.65	.30	1.70
24A	2	31	5.04	.55	.28	1.33	.33	.49	5.86	.08	1.06	1.08	.42	2.26
24A	3	30	5.36	.57	.30	2.10	.46	.51	2.97	.49	1.41	1.84	.62	3.24
24A	5	28	5.71	.63	.22	1.57	.41	.44	3.67	.39	1.92	2.73	.70	3.22
27A	1	28	4.19	.73	.08	1.93	.50	.30	3.76	.22	.54	.46	.55	1.63
27A	3	27	4.99	.71	.10	1.86	.54	.33	4.20	.21	1.10	1.65	.76	2.13
28A	3	28	4.40	1.05	.25	1.27	.44	-.05	3.23	.14	.96	1.11	.74	1.72
32A	1	26	3.53	.49	.44	1.63	.49	.61	1.51	.61	.73	1.15	.67	1.71
34A	1	28	3.40	.74	.07	1.76	.45	.29	2.90	.27	.88	1.12	.75	1.92
34A	2	27	4.03	.60	.26	1.28	.41	.47	2.15	.47	1.09	1.71	.79	1.94
37A	1	34	4.27	1.82	.75	1.85	.36	-.54	4.02	.48	.81	.66	.41	1.72
37A	2	33	5.21	1.23	.39	1.75	.34	-.20	2.86	.17	1.01	1.14	.45	1.86
37A	3	32	5.41	1.23	.39	1.66	.37	-.21	3.34	.13	1.24	1.69	.46	2.01
37A	4	31	5.59	.69	.10	1.56	.39	.11	2.66	.05	1.54	2.15	.60	2.91
37A	6	29	5.98	1.63	.65	1.16	.37	-.45	1.88	.45	2.26	3.47	.57	2.93
37A	7	28	6.11	1.57	.61	1.09	.35	-.42	1.30	.42	2.50	3.83	.42	2.10
37A	8	27	6.25	1.14	.32	1.57	.43	-.13	3.27	.08	2.78	4.08	.32	1.59
38A	2	26	4.84	.63	.21	1.52	.42	.43	4.50	.32	.99	1.01	.59	2.00
38A	3	25	5.33	1.03	.23	1.67	.33	-.03	3.94	.22	1.24	1.72	.57	1.95
38A	5	23	5.40	.76	.04	1.61	.33	.26	4.52	.12	1.69	2.20	.46	1.52
42A	3	28	4.37	.61	.25	1.68	.49	.46	2.61	.45	.85	1.10	.43	1.53
42A	5	27	5.30	1.32	.45	2.14	.42	-.27	4.20	.13	1.32	1.74	.51	2.15
42A	7	25	5.77	1.22	.38	1.86	.41	-.19	3.73	.04	1.84	2.60	.52	2.54
42A	8	24	5.96	1.31	.44	1.28	.34	-.26	.89	.26	2.16	3.17	.51	2.39
44A	2	34	6.07	.77	.04	1.73	.36	.26	4.08	.11	.99	1.21	.27	.90
44A	3	33	6.29	.65	.19	1.22	.34	.41	2.36	.40	1.30	1.86	.46	1.74
44A	5	31	6.61	.94	.15	1.83	.45	.06	4.35	.26	1.84	2.94	.47	1.85
44A	6	30	6.79	1.71	.70	.77	.28	-.49	3.62	.46	2.23	3.68	.46	1.60
44A	7	29	6.99	1.58	.62	1.15	.31	-.43	2.32	.42	2.61	3.96	.34	1.00
44A	8	28	7.09	1.21	.37	1.72	.43	-.19	3.95	.12	2.82	4.09	.38	1.03
45A	1	30	3.81	.58	.28	1.66	.42	.49	5.18	.32	1.24	1.61	.55	1.94
45A	2	29	5.39	1.39	.50	1.55	.27	-.32	4.37	.17	1.22	1.01	.51	1.73
45A	3	28	5.57	1.29	.43	2.01	.28	-.25	4.62	.17	1.34	1.23	.50	1.72
45A	4	27	5.92	1.33	.46	1.78	.27	-.28	4.01	.17	1.64	1.58	.59	1.94
45A	5	26	6.20	1.01	.21	1.53	.27	-.01	3.45	.17	1.87	2.19	.53	1.81
46A	2	30	4.84	.90	.10	1.67	.38	.11	4.84	.31	1.45	1.48	.62	2.89
46A	3	29	5.29	1.48	.56	1.85	.42	-.37	4.91	.15	1.59	1.83	.71	3.23
46A	4	28	5.46	1.25	.40	2.16	.48	-.22	4.93	.27	1.76	2.24	.77	3.47
46A	8	26	5.89	.97	.18	1.38	.30	.03	2.86	.11	2.36	3.54	.36	1.53
48A	2	31	4.30	.30	.92	.91	.39	.84	1.92	.64	.65	.78	.48	2.03
48A	3	30	5.15	1.32	.45	2.22	.37	-.27	4.63	.14	1.02	1.20	.59	2.15
48A	4	29	5.42	1.03	.22	1.73	.36	-.03	3.71	.19	1.25	1.73	.64	2.19
48A	6	27	5.96	1.17	.34	1.54	.34	-.16	3.05	.08	1.82	2.63	.42	1.61
48A	7	26	6.21	.78	.02	1.38	.35	.24	1.90	.24	2.11	2.98	.34	1.24
50A	1	25	3.91	.71	.11	1.59	.39	.33	2.24	.33	.76	.79	.50	.89
51A	1	28	3.14	.63	.21	.87	.29	.43	2.21	.42	.70	1.24	.78	3.89
51A	3	26	3.66	.90	.11	1.66	.47	.10	4.06	.21	1.40	2.08	.73	3.73
51A	4	25	4.22	.46	.51	1.50	.39	.65	5.52	.49	1.63	2.32	.59	2.44
52A	4	25	3.40	.56	.32	1.73	.44	.52	3.72	.49	.93	1.60	.68	2.60
53A	2	26	3.43	.66	.17	1.66	.44	.39	4.12	.31	1.33	1.23	.75	3.25
53A	4	25	3.97	.67	.16	1.25	.40	.38	3.03	.36	1.48	2.06	.72	2.94
54A	1	27	3.04	.43	.55	1.13	.37	.68	2.71	.67	.85	1.39	.50	2.17
54A	2	26	3.50	.49	.44	.94	.36	.61	3.00	.60	1.10	1.68	.82	3.46
56A	1	28	3.40	.53	.37	1.44	.39	.56	2.53	.56	.75	1.32	.48	2.40
56A	2	27	3.83	.61	.24	1.39	.42	.46	1.62	.46	1.09	1.65	.83	3.93
56A	4	25	4.41	.57	.30	1.44	.48	.51	3.04	.49	1.57	2.22	.77	3.49
57A	1	27	3.50	.36	.74	1.30	.37	.77	3.40	.76	.90	1.45	.41	1.86
57A	2	26	3.81	.81	.01	1.98	.44	.20	3.04	.16	1.08	1.36	.53	2.35
57A	3	25	4.36	.45	.51	.92	.33	.66	3.26	.64	1.35	1.49	.60	2.63
58A	1	29	3.43	.49	.44	1.32	.42	.61	2.54	.60	1.23	2.15	.48	2.74
58A	2	28	3.76	.59	.27	1.49	.44	.48	3.27	.45	1.38	2.04	.79	4.36
58A	4	26	4.30	.50	.43	1.12	.38	.60	2.22	.60	1.91	2.80	.83	4.48
58A	5	25	4.54	.89	.10	1.21	.37	.11	1.53	.11	2.12	3.23	.55	3.00
59A	2	30	3.61	.52	.39	1.73	.46	.58	1.40	.57	1.10	1.51	.50	2.79
59A	3	28	3.89	.45	.51	1.11	.39	.66	4.05	.62	1.31	1.73	.74	4.16
59A	5	26	4.28	.72	.09	1.44	.44	.32	2.50	.31	1.78	2.66	.71	3.87
59A	6	25	4.55	.77	.03	1.53	.48	.25	2.47	.24	2.19	3.26	.57	3.05
59A	7	24	4.79	.63	.21	1.40	.45	.43	1.90	.43	2.47	3.99	.30	1.70
61A	1	47	5.15	1.27	.42	2.42	.40	-.24	4.26	.10	.79	.88	.27	1.42
61A	2	46	5.42	1.28	.42	2.39	.43	-.24	4.51	.16	1.06	1.36	.52	2.79
61A	3	45	5.57	1.17	.34	1.81	.39	-.16	2.91	.10	1.29	1.72	.67	3.59
61A	5	43	5.83	1.34	.46	1.44	.33	-.28	1.93	.28	1.95	2.97	.70	4.02
61A	6	42	6.02	1.31	.45	1.58	.36	-.26	2.15	.26	2.39	3.93	.56	3.24
61A	7	41	6.18	1.15	.33	1.63	.36	-.14	2.25	.12	2.72	4.23	.37	2.17
73A	1	48	5.06	1.53	.59	2.45	.41	-.40	4.19	.31	.65	.65	.38	1.80
73A	2	47	5.31	1.22	.38	2.11	.40	-.20	4.02	.11	.94	1.26	.60	2.95
73A	4	45	5.75	1.27	.42	2.11	.38	-.24	3.55	.15	1.58	2.49	.70	3.30
73A	5	44	5.91	1.28	.43	1.53	.32	-.24	.67	.24	1.89	2.68	.62	3.01
73A	6	43	6.13	1.10	.28	1.86	.34	-.09	3.22	.11	2.30	3.66	.40	1.84
74A	2	48	5.40	1.14	.32	1.71	.36	-.13	2.97	.03	.73	.88	.61	3.61
74A	3	47	5.55	1.13	.31	1.76	.37	-.12	3.06	.05	1.00	1.39	.68	4.05
74A	5	45	5.91	1.10	.29	1.87	.34	-.10	2.74	.04	1.78	2.77	.69	3.95
74A	6	44	6.02	1.01	.21	1.67	.33	-.01	2.27	.07	2.06	3.28	.70	3.91
74A	7	43	6.22	.96	.16	1.37	.28	.04	1.94	.03	2.51	3.91	.47	2.56
75A	4	28	5.33	1.02	.22	2.10	.38	-.02	4.24	.25	2.06	3.11	.68	4.42
75A	5	27	5.51	1.20	.36	1.83	.38	-.18	3.38	.08	2.26	3.20	.69	4.51
75A	7	25	5.92	.80	.00	1.87	.32	.22	2.29	.21	2.80	3.86	.57	3.34
76A	1	34	4.54	.34	.79	1.31	.36	.79	3.47	.77	1.04	1.05	.25	1.56
76A	2	33	4.74	.36	.74	1.14	.44	.77	3.05	.76	1.19	1.24	.42	2.60
76A	3	32	4.86	.58	.29	1.33	.42	.50	2.66	.49	1.39	1.87	.46	2.84
76A	4	31	5.28	1.00	.20	2.48	.41	.00	4.31	.26	1.74	2.46	.51	3.06

Table 3.9.4 (continued)

c. APC's.

76A	6	29	5.62	1.64	.66	1.48	.34	-.46	3.56	.42	2.31	3.41	.56	3.67
76A	7	28	5.77	1.18	.25	2.02	.41	-.17	4.03	.18	2.59	2.90	.69	3.20
75A	8	27	5.67	1.22	.20	1.69	.37	-.21	3.11	.18	2.95	2.30	.60	2.58
76A	9	26	6.21	1.55	.60	1.16	.28	-.41	2.04	.41	3.28	4.51	.37	2.16
78A	1	33	5.18	1.12	.30	1.61	.37	-.11	1.15	.11	1.52	1.70	.56	2.30
79A	3	32	5.40	1.31	.44	1.70	.43	-.26	1.96	.26	1.66	.36	.56	2.21
78A	4	31	4.61	.69	.47	1.91	.31	.63	3.04	.62	1.85	1.21	.59	2.71
79A	1	31	4.25	.57	.31	1.10	.38	.51	1.85	.51	1.21	1.28	.59	2.78
79A	2	30	4.49	.57	.48	.95	.38	.64	1.68	.64	1.39	1.54	.78	3.93
79A	3	30	3.16	.70	.12	1.44	.36	.35	4.05	.26	1.26	1.95	.50	2.59
79A	4	29	4.67	.69	.13	1.71	.47	.35	4.41	.22	1.62	2.01	.82	4.10
79A	5	29	3.50	.69	.45	1.21	.34	.62	3.50	.59	1.45	2.82	.54	2.50
79A	6	25	4.92	.09	.13	1.65	.48	.35	3.96	.27	1.97	2.57	.58	3.24
79A	7	28	3.79	.75	.02	1.61	.39	.26	3.25	.19	1.68	3.07	.57	2.69
79A	8	27	5.57	1.02	.22	2.12	.35	-.02	4.68	.31	2.22	3.01	.45	2.18
79A	10	25	6.11	1.08	.27	1.75	.28	-.07	2.39	.16	2.95	3.07	.36	1.22
80A	1	30	4.75	.38	.66	1.28	.40	-.74	4.31	.69	1.17	1.40	.33	1.65
80A	2	29	5.10	.62	.02	2.29	.41	.19	4.83	.27	1.62	2.01	.45	2.54
80A	3	25	5.29	.96	.17	2.25	.42	.04	4.92	.34	1.67	2.37	.63	3.40
80A	5	26	5.77	.97	.17	1.91	.33	.03	3.25	.15	2.34	3.65	.50	2.90
80A	6	25	6.05	1.32	.45	1.83	.38	-.27	3.32	.25	2.59	3.41	.53	2.32
80A	7	24	6.23	1.18	.35	1.46	.34	-.16	.68	.16	2.78	3.64	.40	1.67
86A	1	28	3.76	1.45	.56	1.57	.40	-.30	.96	.36	4.01	5.59	.69	9.47
86A	2	27	3.83	1.60	.75	.90	.28	-.35	5.09	.35	4.26	5.99	.62	6.46
86A	3	28	3.87	1.01	.21	.90	.26	-.01	5.96	.50	4.53	6.02	.63	6.57
86A	4	25	3.89	1.15	.53	.68	.21	-.14	7.89	.67	4.68	6.26	.61	8.33
86A	5	25	3.18	.74	.07	1.46	.40	.30	2.93	.27	1.29	2.00	.65	3.05
86A	7	24	3.59	1.15	.33	.65	.21	-.14	7.89	.67	4.63	6.24	.61	8.33
86A	9	24	3.61	.74	.07	1.85	.44	-.29	1.29	.29	1.65	2.55	.56	2.52
90A	1	34	3.63	1.92	.81	2.07	.43	-.57	1.21	.57	1.39	4.30	.40	2.50
90A	3	32	5.08	1.40	.20	2.43	.33	.00	3.15	.16	1.79	1.61	.43	2.77
90A	4	31	5.63	1.25	.40	2.07	.30	-.22	1.49	.22	1.59	1.52	.45	2.93
90A	5	30	5.71	1.26	.41	1.91	.42	-.22	1.66	.22	2.13	2.20	.54	3.48
90A	7	28	5.96	1.06	.25	1.26	.26	-.06	4.32	.26	2.50	3.17	.62	3.97
90A	8	27	6.13	1.08	.27	1.87	.38	-.07	2.81	.08	2.82	3.87	.58	3.75
90A	9	26	6.24	1.16	.33	1.35	.32	-.14	2.54	.11	3.05	4.24	.45	2.88
90A	10	25	6.40	1.08	.26	1.32	.29	-.06	1.86	.20	1.35	4.81	.32	1.91
91A	1	34	5.15	.85	.05	1.84	.33	.16	1.81	.15	1.25	.80	.38	1.72
91A	2	33	5.34	1.18	.33	1.95	.38	-.14	1.90	.13	1.43	1.57	.45	3.21
91A	4	31	5.61	1.29	.43	1.65	.30	-.25	1.98	.24	2.02	2.66	.60	4.23
91A	6	29	5.93	1.08	.27	1.47	.35	-.08	3.88	.06	2.76	3.23	.64	4.21
91A	7	28	6.02	1.00	.20	1.45	.34	.00	2.06	.06	2.98	3.40	.65	4.14
91A	8	27	6.16	1.21	.39	1.76	.38	-.21	.09	.21	3.29	3.97	.51	3.06
91A	9	26	6.28	.57	.17	2.18	.25	.03	3.00	.12	3.66	4.21	.36	2.06
93A	2	34	5.50	.49	.20	2.11	.41	.01	2.38	.08	1.71	1.89	.48	1.68
93A	3	33	5.30	.64	.20	1.82	.22	.42	2.50	.41	1.53	2.11	.28	1.96
93A	4	32	5.73	.86	.09	1.93	.23	.15	3.23	.06	2.89	.87	.41	3.12
93A	5	31	5.83	.99	.17	1.89	.37	.19	3.11	.30	2.97	.59	.67	3.59
93A	6	30	5.92	.75	.10	1.39	.29	.32	2.95	.23	3.06	1.16	.55	4.60
93A	8	28	6.09	.85	.05	1.25	.27	.16	4.39	.22	3.37	2.41	.54	4.28
93A	9	27	6.15	.62	.24	1.05	.26	.45	4.29	.37	3.56	2.86	.50	4.02
93A	10	26	6.31	.91	.11	1.77	.32	.10	2.81	.05	4.00	3.44	.41	3.16
94A	1	33	5.03	.82	.02	1.76	.29	.19	3.40	.36	1.34	1.02	.61	2.21
94A	2	32	5.26	1.09	.28	1.67	.30	-.09	4.47	.27	1.50	1.35	.58	3.19
94A	3	31	5.45	.83	.03	1.86	.33	.19	2.65	.16	1.74	1.93	.60	3.56
94A	4	30	5.59	1.01	.21	1.85	.37	-.01	2.93	.12	1.97	2.29	.69	4.10
94A	5	29	5.73	.69	.13	1.07	.33	.35	4.85	.17	2.26	2.64	.67	4.19
94A	7	27	6.03	1.05	.24	1.80	.43	-.05	2.36	.06	2.94	4.41	.44	2.79
94A	8	26	6.16	1.48	.50	1.38	.36	-.37	2.31	.36	3.25	4.65	.28	1.78
96A	1	32	5.66	.91	.01	2.11	.35	.21	3.60	.10	2.05	.35	.60	3.76
96A	3	30	5.94	.74	.08	1.74	.32	.30	3.31	.25	2.54	1.35	.63	4.02
96A	4	29	6.12	.68	.09	1.79	.36	.12	3.16	.07	2.88	1.88	.51	3.66
96A	5	28	6.24	.62	.22	1.81	.33	.44	3.88	.38	3.16	2.19	.44	2.87
96A	6	27	6.53	.43	.38	1.56	.34	.69	2.20	.68	3.65	2.10	.65	2.51
97A	1	31	5.27	.75	.02	1.90	.31	.24	3.44	.17	2.19	2.11	.33	2.41
97A	2	30	5.56	.65	.05	1.72	.30	.16	2.29	.15	2.40	2.32	.48	3.48
97A	3	29	5.71	1.03	.22	1.60	.31	-.03	2.41	.08	2.54	2.49	.59	4.25
97A	4	28	5.89	.79	.19	1.95	.32	.04	3.12	.13	2.76	3.12	.60	4.29
97A	5	27	6.02	.81	.01	1.68	.29	.20	1.51	.20	2.95	3.85	.57	3.65
97A	7	25	6.23	1.20	.36	1.62	.32	-.18	1.74	.17	3.43	4.58	.59	3.85
97A	8	24	6.33	1.12	.31	1.07	.28	-.12	3.43	.12	3.69	5.12	.51	3.21
98A	1	28	5.73	.95	.09	1.25	.28	-.19	1.71	.11	2.52	.53	.60	4.29
98A	2	27	5.83	1.32	.45	1.72	.26	-.27	1.67	.27	2.80	1.45	.58	4.16
98A	3	26	5.91	.58	.29	1.19	.43	.49	1.81	.49	3.08	2.22	.62	4.39
98A	4	25	6.00	1.03	.23	1.93	.59	-.03	3.89	.21	3.39	2.96	.57	4.00
101A	1	48	5.31	1.48	.56	2.40	.34	-.38	1.36	.37	.64	.59	.65	2.62
101A	2	47	5.01	1.37	.49	1.97	.33	-.30	.98	.30	.83	1.04	.44	3.16
101A	3	46	5.03	1.08	.27	1.47	.30	-.08	1.46	.07	1.17	1.49	.50	3.39
101A	4	45	5.05	1.01	.21	1.17	.39	-.01	3.27	.15	1.41	1.57	.61	4.63
102A	1	43	5.16	.38	.43	.88	.28	.62	3.53	.45	.56	.73	.62	1.87
102A	3	46	5.62	.39	.45	.75	.28	.61	3.03	.53	1.20	1.79	.52	1.88
102A	4	45	5.21	.38	.24	.64	.27	.50	7.04	.50	1.52	2.05	.61	1.57
102A	5	44	5.07	.57	.31	1.17	.18	.51	1.68	.51	1.82	2.30	.40	1.46
111A	2	30	5.17	1.03	.22	2.22	.40	-.03	4.44	.23	1.88	1.21	.91	4.07
111A	4	28	5.56	1.02	.22	2.07	.36	-.02	3.40	.15	2.31	3.00	.70	4.71
111A	5	27	5.74	.90	.10	1.79	.35	.11	2.68	.04	2.55	3.45	.72	4.90
111A	6	26	5.87	1.21	.17	1.66	.37	-.19	2.22	.18	2.79	4.04	.63	4.47
111A	7	25	5.95	.98	.16	1.31	.34	.02	1.95	.05	2.99	4.37	.55	3.62
111A	8	24	6.05	.94	.16	1.30	.44	.06	.48	.06	3.26	4.10	.59	4.10
111A	9	23	6.21	.99	.20	1.65	.45	.01	2.50	.09	3.65	5.24	.59	2.81
112A	3	30	5.00	.68	.16	2.18	.30	.36	4.69	.19	2.25	1.23	.41	3.43
112A	4	29	5.39	1.30	.48	2.00	.32	-.30	2.88	.28	2.51	1.85	.58	4.61
112A	6	27	5.72	1.58	.62	1.40	.28	-.43	1.89	.43	2.97	3.01	.61	4.81
112A	7	26	5.85	1.14	.32	1.53	.30	-.13	1.43	.13	3.19	3.53	.62	4.59
112A	8	25	5.93	1.48	.56	1.38	.32	-.38	1.39	.37	3.30	4.06	.60	4.76
112A	9	24	6.02	1.05	.24	1.78	.37	-.05	3.60	.18	3.60	4.59	.50	4.04
113A	2	29	5.33	.52	.12	2.36	.37	.09	3.75	.18	1.67	2.01	.61	3.99
113A	3	28	5.47	1.28	.42	1.78	.36	-.26	1.19	.26	1.87	2.65	.58	3.90
113A	4	27	5.66	.92	.13	1.52	.31	.08	.59	.09	2.23	3.29	.56	3.93
113A	5	26	5.85	.99	.19	1.67	.29	.01	2.60	.10	2.55	3.82	.54	3.96
113A	6	25	5.97	1.09	.28	.93	.30	-.09	4.36	.28	2.79	3.50	.61	4.45
113A	8	23	6.13	.94	.14	1.34	.46	.06	1.97	.03	3.22	4.35	.61	4.31
114A	1	32	4.55	.99	.19	2.14	.30	.01	2.58	.09	1.14	1.03	.66	1.77
114A	3	31	5.48	1.10	.29	1.85	.22	-.09	1.37	.09	1.25	1.01	.56	1.81
114A	5	30	5.78	1.52	.59	1.62	.21	-.40	1.77	.40	1.55	1.53	.56	2.01
115A	3	32	4.93	.87	.07	2.29	.39	.14	4.78	.29	1.39	1.64	.67	3.17
115A	4	31	5.18	1.02	.22	2.14	.38	-.02	4.11	.24	1.65	1.89	.61	4.00
115A	5	30	5.42	1.15	.33	1.92	.38	-.16	3.49	.10	1.91	2.32	.56	3.86
115A	7	28	5.74	1.07	.20	1.69	.32	-.07	2.44	.05	2.37	3.60	.61	4.22
115A	8	27	5.87	1.00	.20	1.33	.32	.00	1.53	.07	2.66	3.92	.57	3.86
115A	9	26	6.00	1.59	.61	1.19	.34	-.43	1.46	.43	2.95	4.46	.57	3.91
115A	10	25	6.09	1.25	.40	1.19	.41	-.22	1.20	.21	3.19	4.33	.63	4.29
115A	11	24	6.19	1.20	.39	1.23	.40	-.18	1.45	.18	3.50	5.17	.65	2.98
122A	1	25	3.83	.96	.17	1.21	.33	.04	3.34	.15	.60	1.09	.51	1.65
122A	3	23	4.74	.81	.01	1.51	.45	.21	3.89	.04	1.11	1.29	.56	1.60
123A	1	29	3.37	.87	.07	1.55	.39	.14	5.04	.33	.56	.67	.46	2.67
123A	2	28	3.71	.70	.11	1.53	.37	.34	3.66	.25	.81	.96	.53	3.00
123A	3	27	3.99	.55	.34	1.33	.37	.54	3.01	.52	1.12	1.36	.65	3.43
123A	4	26	4.44	.59	.28	1.58	.43	.49	3.54	.45	1.51	2.22	.87	4.13
123A	6	24	4.88	.65	.19	1.20	.38	.41	1.13	.41	1.96	2.78	.67	3.00
125A	1	27	3.91	.66	.17	1.58	.43	.39	4.49	.27	.94	.74	.58	2.91
125A	3	25	4.45	.53	.38	1.40	.41	.56	3.69	.53	1.52	2.02	.70	3.13
125A	4	24	4.84	.65	.19	1.31	.37	.40	2.68	.39	1.86	2.53	.63	3.02
127A	1	27	4.22	.87	.07	1.65	.43	.16	4.34	.23	1.98	2.86	.87	3.60
127A	3	25	4.38	.69	.14	.96	.34	.36	.97	.36	2.60	4.19	.72	3.87
127A	4	24	4.80	.98	.14	1.77	.50	.36	4.52	.22	2.87	4.31	.46	3.00
129A	1	29	4.17	.70	.12	1.07	.31	.35	1.09	.35	1.14	1.78	.87	2.37
129A	3	27	4.76	.80	.00	1.66	.46	.22	4.13	.10	1.63	2.27	.85	2.85
129A	4	26	5.14	.68	.16	1.37	.39	.37	3.87	.30	1.80	2.60	.52	2.23
130A	1	25	3.95	.94	.14	1.26	.39	.08	3.77	.09	1.09	.96	.71	2.97
130A	3	23	4.69	.74	.07	1.19	.38	.30	3.02	.27	1.54	2.10	.70	2.66

Table 3.9.4 (continued)

WINDOW	CC#	THRESH	LOG A	H/W	NOISE (H/W)*	H*W/A	(H*W)/P	DIFF	SK_EW	ASYM	SDEV	GL DIFF	EEP	EP
1T	12	35	5:57	.35	.76	1:50	.28	.75	1.95	.73	3.96	1:09	.32	1:21
1T	19	24	6:03	.40	.64	1:55	.27	.73	1.70	.74	3.27	1:06	.35	2.05
11T	8	26	5:19	1.09	.25	1:49	.29	-.09	1.52	.08	1.07	.87	.30	.77
11T	9	32	5:21	.75	.90	1:37	.32	.28	1.93	.27	1:29	1:56	.38	1:48
11T	10	32	5:02	.46	.51	1:34	.44	.05	2.78	.64	5.07	6:58	.30	2.04
3R	8	27	6:02	.45	.52	1:58	.27	.06	2.05	.59	2.56	.49	.40	1.50
3R	10	26	6:03	.42	.56	1:58	.27	.10	2.05	.60	2.65	1:20	.39	1.56
20R	4	25	5:09	.24	.07	1:20	.35	.29	3.56	.83	.93	.98	.58	1:06
31R	4	29	5:37	.59	.28	2:09	.34	.49	4.19	.42	1.28	1:89	.41	.99
41R	5	33	5:26	1.32	.65	1:40	.18	-.27	2.89	.96	1:15	.69	.29	1:12
41R	8	28	6:45	.48	.08	1:44	.35	.13	4.04	.10	2:32	1:03	.46	2:21
41R	9	27	6:01	.90	.11	1:28	.46	.10	3.11	.09	2:03	1:02	.38	1:08
41R	10	26	6:13	1.18	.35	1:09	.42	.18	3.53	.06	2:00	4:23	.41	8:01
41R	11	25	6:05	1.07	.26	1:06	.35	.07	3.04	.10	3:20	4:05	.36	1:40
57R	4	26	4:47	.91	.24	1:71	.35	.40	2.46	.45	1:39	1:21	.56	1:34
58R	5	24	4:47	1.37	.61	1:82	.41	.42	2.25	.42	1:20	.83	.44	1:19
59R	8	32	5:06	1:40	.81	1:88	.42	.33	3.37	.89	1:28	.83	.30	1:02
63T	3	24	5:06	2.31	.89	1:88	.40	.68	3.33	.88	.95	.82	.36	1:02
68T	4	26	4:34	.58	.67	1:76	.42	.74	2.77	.74	.73	.41	.47	1:03
69T	6	26	4:33	.42	.59	1:52	.38	.70	2.07	.70	.55	1:37	.52	1:01
1T	16	25	3:28	.79	.01	1:98	.42	.23	4.84	.24	1:45	1:86	.40	1:03
3T	6	30	3:18	.96	.13	1:64	.44	.35	4.81	.88	1:49	1:30	.80	3:25
3T	8	29	3:38	.73	.10	1:56	.41	.35	4.51	.91	1:75	2:38	.58	2:02
3T	12	26	3:09	.43	.14	1:48	.41	.09	4.91	.10	3:35	9:28	.83	5:26
4DT	5	23	4:13	.99	.19	1:61	.38	.01	3.87	.12	.81	1:03	.52	1:07
5BT	2	32	3:31	.39	.06	1:58	.40	.74	3.63	.73	.65	.37	.59	.89
3R	2	30	3:33	1.05	.24	1:41	.41	-.04	3.67	.06	1:04	1:14	.76	4:10
3R	3	30	3:18	.64	.54	1:58	.23	.07	4.38	.92	1:38	1:07	.78	4:38
3R	4	29	3:09	1.46	.55	1:58	.42	.36	3.76	.30	1:09	2:73	.59	3:10
3R	5	30	3:40	.37	.05	1:34	.26	.76	4.06	.95	1:91	6:25	.47	6:04
3R	6	28	3:37	1.01	.23	1:15	.34	.53	4.23	.37	2:15	2:56	.47	2:41
3R	7	28	3:30	1:00	.67	1:22	.49	.47	3.30	.57	2:09	2:02	.70	4:09
3R	9	29	3:31	1:09	.28	1:38	.47	.30	.89	.50	5:60	5:96	.56	3:30
9R	3	26	3:21	.47	.49	1:10	.33	.06	2.36	.83	3.95	.74	.68	2:12
9R	4	30	3:09	.53	.36	1:35	.39	.57	4.67	.40	.55	1:76	.71	1:33
9R	7	29	3:30	.64	.23	1:08	.37	.46	5.09	.44	.75	1:42	.66	.86
9R	8	25	3:33	.94	.15	1:02	.41	.06	4.37	.26	1.06	.78	.60	1:23
9R	12	24	3:25	1.25	.40	1:01	.43	-.22	4.95	.27	.80	.85	.64	1:04
35R	3	24	3:06	.86	.17	1:06	.39	.58	4.66	.58	.82	.80	.68	1:04
37R	2	27	4:39	.78	.03	2:13	.42	.25	.91	.23	.80	.93	.48	1:53
37R	10	24	4:17	.90	.12	1:51	.39	.35	1.48	.35	.59	.38	.80	1:03
39R	3	24	4:49	1:43	.56	1:78	.40	.30	4.71	.34	1:11	.39	.43	1:04
62T	5	24	4:47	2:16	.62	1:66	.41	.65	2.81	.84	1.01	.24	.42	1:30
97A	9	24	3:09	1.43	.76	1:89	.45	.54	1.80	.84	.95	.95	.86	1:21
2N	1	32	3:10	.31	.41	1:84	.48	.59	1.80	.84	1.20	1:81	.82	2:20
2N	2	31	3:47	.37	.71	1:38	.44	.76	1.05	.76	1.58	1.95	.80	2:56
2N	3	31	3:39	.34	.78	1:05	.36	.79	3.66	.79	.69	.70	.82	2:12
2N	5	30	4:26	.73	.08	1:76	.34	.31	1.15	.27	.70	.38	.55	1:02
2N	6	30	3:56	.58	.36	2:59	.30	.56	4.38	.49	.86	1.02	.88	1:04
2N	7	30	3:30	.90	.10	1:55	.34	.11	6.50	.35	.92	.23	.65	1:56
2N	9	29	4:09	.46	.50	1:93	.41	.05	6.23	.60	1.09	.58	.64	1:32
2N	10	29	3:28	.97	.18	1:27	.34	.03	5.48	.62	.89	.98	.69	1:16
2N	11	28	6:80	1.11	.29	1:80	.38	-.10	5.02	.36	1.69	.26	.40	.79
2N	12	28	4:03	.52	.38	1:67	.39	.57	3.09	.55	.98	1:18	.62	1:41
2N	13	28	5:61	.34	.79	1:58	.38	.79	1.92	.79	.68	1:50	.58	.75
8N	1	32	5:35	.84	.30	1:99	.32	.62	2.65	.41	1.07	.30	.59	1:15
8N	2	31	5:07	1.01	.21	1:72	.24	-.01	2.74	.11	1.17	.80	.46	.89

Table 3.9.4 (continued)

d. Non-target windows.

1. $20 \leq N \leq 1000$

2. Gray level difference ≥ 0.2

3. $E\&P \geq 0.2$

a. Superslice decision thresholds. Candidate object regions must satisfy all of these conditions.

1. $(h/w)' \leq 1.0$

2. $(h*w)/A \leq 2.5$

3. $|SDEV-3| \leq 2.5$

4. $(E\&P \geq 0.4) \vee ((E\&P \geq 0.25) \wedge (\log_e N > 4.5))$

5. $E_p \geq 0.75$

6. $Max(h,w) < 64$

b. Preclassifier decision thresholds. Candidate target regions must satisfy all of these conditions.

Table 3.9.5. Classification decision threshold. (Feature variables are defined in Table 3.8.1.)

Feature factors		Noise coefficients	Target coefficients
c3		.298+02	.251+03
c4		.858+02	.381+03
c5		.217+01	-.253+02
c7		.675+02	.254+02
c10		.887+02	.294+03
c3	c3	-.218+02	-.118+03
c4	c4	-.884+03	-.132+04
c5	c5	-.466+01	-.180+02
c7	c7	-.334+02	-.349+02
c10	c10	-.719+02	-.106+03
c3	c4	.233+03	.435+03
c3	c5	.789+01	-.200+02
c3	c7	-.341+02	-.287+02
c4	c10	-.624+02	-.121+03
c4	c5	-.697+02	.142+03
c4	c7	.129+03	.783+02
c4	c10	.313+03	.852+02
c5	c7	.111+02	.266+02
c5	c10	.151+02	-.678+01
c7	c10	-.594+02	-.476+01
		-.785+02	-.357+03

Table 3.9.5c. Maximum-likelihood discriminant used at small node using features c3, c4, c5, c7, c10. A region is small if $\log_e N \leq 4.5$.

Feature factors		Noise coefficients	Truck-like coefficients	Tank coefficients
c2		.169+03	.277+02	.179+02
c3		.519-01	.174+02	.698+02
c5		-.551+01	.216+01	.138+02
c6		.632+02	.262+01	.913+01
c10		.259+03	.683+02	.910+02
c11		.157+03	-.132+01	-.189+01
c2	c2	-.325+02	-.160+02	-.273+02
c3	c3	-.112+02	-.658+01	-.176+02
c5	c5	-.454+01	-.520+01	-.132+02
c6	c6	-.500+01	-.339+00	-.457+00
c10	c10	-.153+03	-.582+02	-.473+02
c11	c11	-.264+02	-.688+00	-.367+00
c2	c3	.526+01	-.857+01	-.878+01
c2	c5	-.255+01	.151+01	.222+02
c2	c6	-.211+02	-.892-01	-.951+00
c2	c10	-.808+02	-.203+02	-.119+02
c2	c11	-.487+02	.248+01	.323+01
c3	c5	.283+01	-.303+01	.157+01
c3	c6	.175+01	.835+00	-.287+01
c3	c10	.377+02	-.220+01	-.199+02
c3	c11	.700+01	.125+01	.558+00
c5	c6	-.230+01	.387+00	-.942+00
c5	c10	.127+02	.647+01	-.124+02
c5	c11	.266+01	-.641+00	-.545+00
c6	c10	-.205+02	-.155+01	-.368+01
c6	c11	-.184+02	-.236+00	.150+00
c10	c11	-.774+02	.597+01	.424+01
		-.267+03	-.413+02	-.924+02

Table 3.9.5d. Maximum-likelihood discriminant used at large node using features c2, c3, c5, c6, c10, c11. A region is large if $\log_e N > 4.5$.

Feature	Fisher direction
(h*w)/A	-.45
(h+w)/P	.088
skewness	.028
asymmetry	.56
E&P	-.69

Threshold = -.7736

Table 3.9.5e. Fisher linear discriminant used at truck-like node using features c3, c4, c6, c7, c10.

3.9.5 Classification and context

Our approach to the target cueing problem has been to extract and classify object regions independently of one another. That is, segmentation is based on the assumption that the object regions are individually thresholdable, though not necessarily by the same threshold. Classification is based on information derived from measurements on the individual components but does not take into account the intra- and inter-frame context of a region.

The Gestalt laws of grouping (see [26]) are of interest in this respect since they refer to factors that cause some parts to be seen as belonging more closely together than others. These rules are applications of the basic principles of similarity which assert that region association is partly defined by region resemblance.

There are several types of similarity which could be used with FLIR imagery, e.g., similarity of appearance (size, shape, brightness, etc.), similarity of location or proximity, similarity of spatial arrangement, and temporal similarity (multiple views of the same object in different frames).

Whenever one can confidently group a set of N objects as being similar (based one or more of the types of context discussed above), it may be advantageous to classify them collectively (The Compound Decision problem) rather than independently (The Simple Decision problem).

The compound decision problem can be stated briefly as follows:

There are a set of states of nature $\Omega = \{1,2,\ldots\gamma\}$ and a set of actions $A = \{1,2,\ldots s\}$, associated with an rxs loss matrix L_{ij} being defined for every $i \in \Omega$ and $j \in A$. When the same decision problem is confronted N times, there exists a vector $\vec{\theta}_N = \{\theta_1,\theta_2,\ldots\theta_N\}$ of states of nature where $\theta \in \Omega^N$ and a corresponding vector $X_N = \{x_1,x_2,\ldots x_N\}$ of random variables. θ_k denotes the state in the kth problem and the distribution of x_k is $P(x_k|\theta_k)$. For a given θ_k, x_k is independent of other x's and θ's. In other words

$$P(\vec{x}_N|\vec{\theta}_N) = \prod_{j=1}^{N} P(x_j|\theta_j)$$

We do not assume that the θ's are necessarily independent.

The loss in the compound decision problem is taken to be the average of the losses incurred at each of the N decisions and the compound risk is defined correspondingly.

If all the observations $\vec{X}_N$ are at hand before the individual decisions must be made, one can use a compound decision rule $\vec{t}_N = \{t_1,t_2,\ldots t_N\}$ where $t_k = t_k(j|\vec{X}_N)$ for each $\vec{X}_N$ is a distribution over A, according to which the kth action is chosen. Also one can define a sequential compound decision rule if only the observations $\vec{x}_k$ are at hand before the kth decision is made.

It is possible to work out a decision procedure which is compound Bayes against the distribution $G(\vec{\theta}N)$ where $\theta_N \in \Omega^N$ (for the details see Abend [27]).

The continuation of this project will be concerned with the use of context in general and with the compound decision rule in particular as a way of combining related observations. Naturally, this will require a data base which is sufficiently structured to provide the necessary context. However, a recasting of the problem makes another type of context available.

Consider a set of nested regions (exemplars) produced by the Superslice algorithm. We wish to investigate how these regions can be treated in ensemble as defining (perhaps) a target region. This suggests the following experiment: Given a set of object regions generated by Superslice, classify them independently. Choose a nested region of significance: namely, a subtree in the containment forest (corresponding to a given window or frame) all of whose paths from the root to the terminal nodes are of length $\geq nt$ where $0 \leq t \leq 1$ and n is the number of thresholds used by Superslice. This insures that for a proper choice of t we only consider nested regions which keep appearing for a large fraction of the total number of thresholds.

For each such nested region (subtree), suppose that there is a class, say, w (tank, APC, truck or noise) such that M of the N objects in the subtree have been assigned

to w and $M \geq t\ell$ where ℓ is the length of the longest path in the subtree. (This rule insures that for a proper choice of t the chosen class w really dominates the entire subtree.) We then assign the class w to all of the N objects in the subtree. Otherwise we leave the individual classifications unaltered.

In an experiment using the NVL data base, 315 objects generated by Superslice from 52 windows were considered. The objects were hand picked to belong to the apriori classes tank, APC, truck and noise, and were then classified into five classes, viz. Tank, APC, Truck, Small target, and Noise. The corresponding confusion matrix is shown in Table 3.9.6 a.

We then applied the majority logic context rule on all the containment forests (52 of them) for t = .5; the resulting confusion matrix is shown in Table 3.9.6 b.

A comparison between the two matrices shows an improvement in the false dismissal rate. The false alarm rate is left unchanged, since no significant nested regions (for t = .5) could be found where the noise class dominated the target class. Within the target classes we find a marked improvement in the self-classification of tanks and APC's. However, more trucks in the second case have been misclassified into APC's. This is presumably not due to an error in the majority logic rule, but rather due to the inability of the classifier to discriminate trucks from APC's.

A Priori	Classified as: Tank	APC	Truck	Small Target	Noise
Tank	28	1	2	4	19
APC	10	26	15	35	22
Truck	6	10	10	27	23
Noise	6	1	1	7	62

Table 3.9.6a. Independent classification confusion matrix

A Priori	Classified as: Tank	APC	Truck	Small Target	Noise
Tank	40	1	0	0	13
APC	13	38	11	30	16
Truck	6	15	6	27	22
Noise	6	1	1	7	62

Table 3.9.6b. Majority logic classification confusion matrix

The majority logic context rule is not necessarily a superior classification procedure, since Superslice considers only the best exemplars and may therefore produce a better classification. However, the present study does support the relevance of low-level context for classification validation.

3.10 The Dynamic Environment

The work described heretofore has considered the analysis of single frames. However, inasmuch as the sensor is capable of generating 30 frames per second and the hardware is capable of analyzing about 3 frames per second, it is worthwhile to investigate how information culled from sequences of frames can improve the performance of the system. There are two ways in which sequence data can be helpful. First, a high scanning rate allows a succession of views of the same scene with only a small amount of change (dependent on platform motion). Thus, image statistics should be relatively stable and multiple measurements may allow a reduction of the standard deviation of feature values. Second, the use of motion information can provide a better description of the object regions in a scene. For this project only a small data base of ten sequential frames was available (Figure 3.10.1). The image content and quality are similar to those of the NVL data base. The sequence corresponds to every other frame from the FLIR sensor over a span of 2/3 of a second. The images show a tank against a background of trees, and fade away more with each frame. While this data base was not large enough to permit meaningful tests, it did allow some exploratory work.

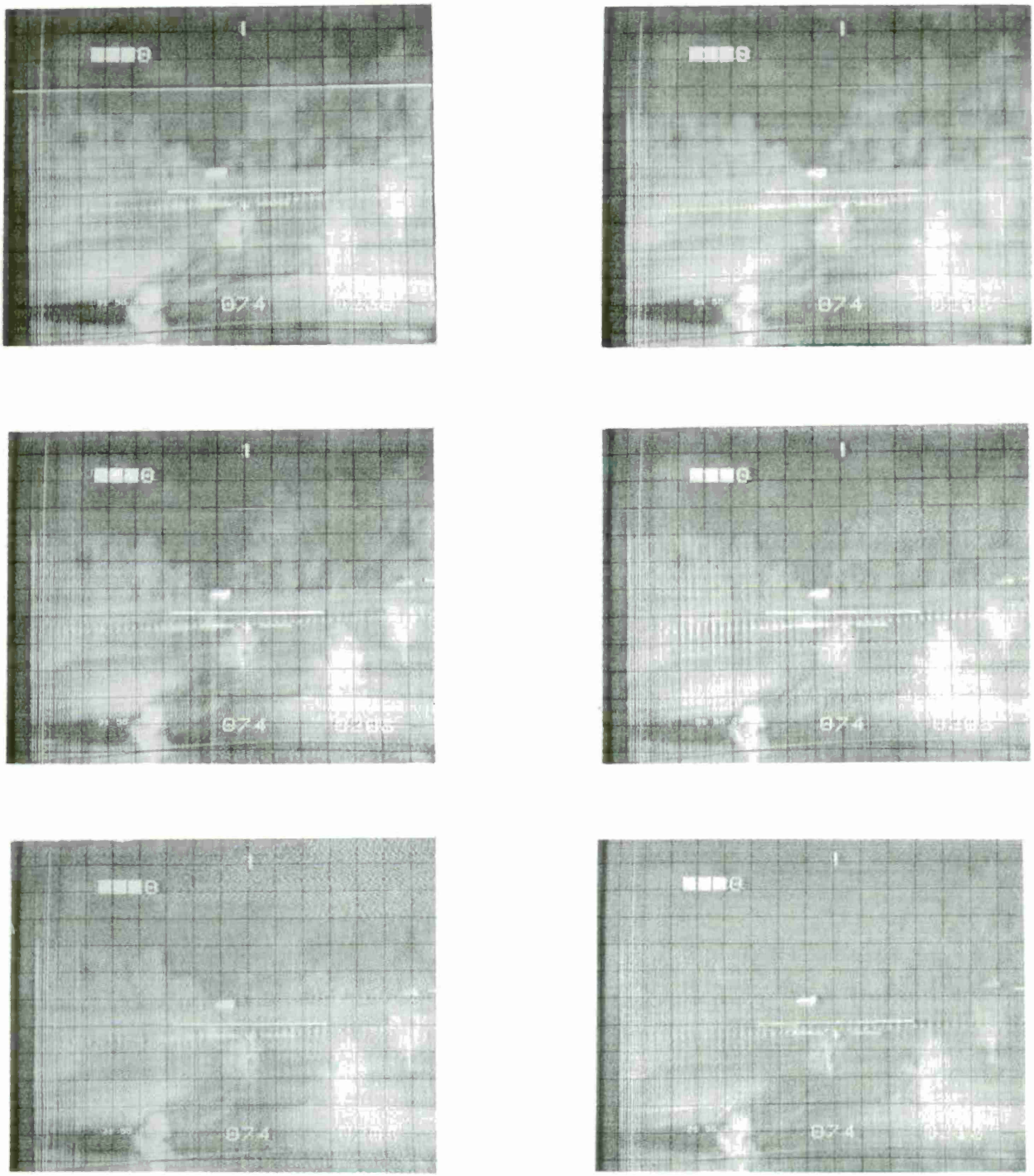


Figure 3.10.1. Ten sequential frames.

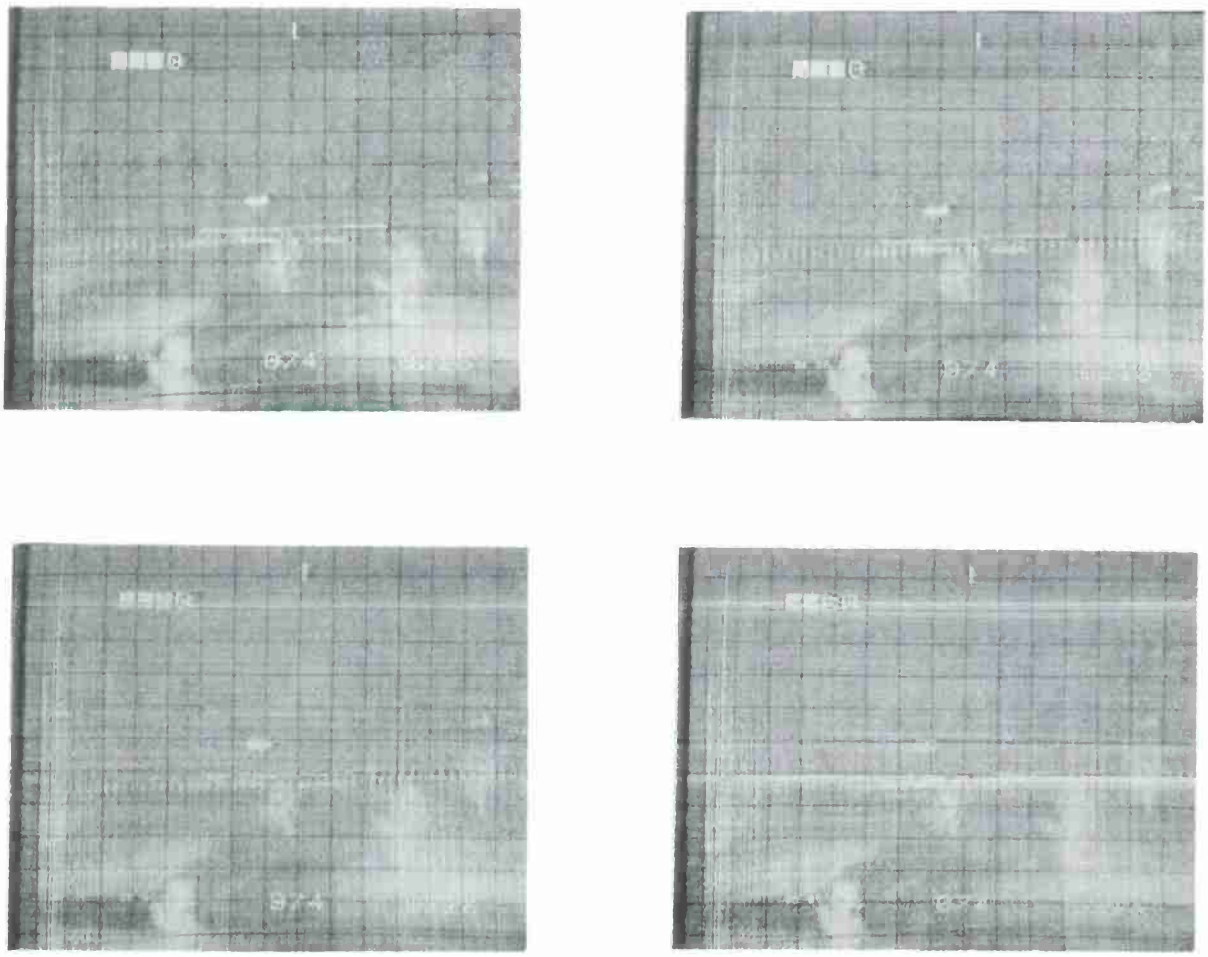
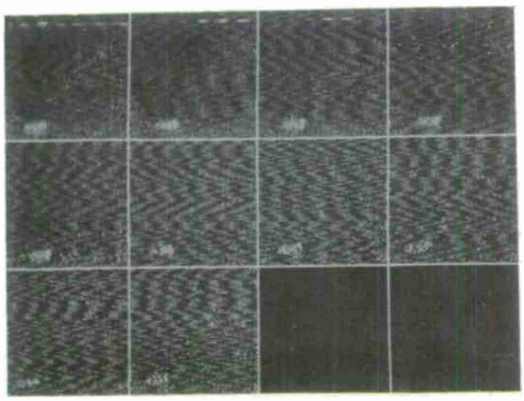


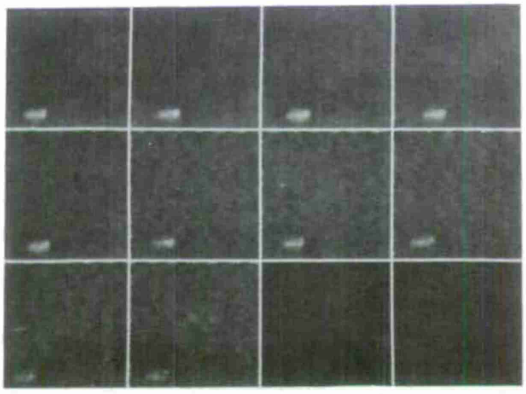
Figure 3.10.1 (continued)

3.10.1 Threshold selection

One does not expect (time-) adjacent frames to differ radically and therefore it should be possible to use good thresholds from the previous frame to segment the current frame or at least to guide the selection of thresholds in the current frame. A sequence of 10 windows was extracted and smoothed (Figure 3.10.2) and a best threshold was chosen for each. Figure 3.10.3 shows the effect of choosing a lower threshold or a higher threshold. As may be noted, the adjacent thresholds have a fairly negligible effect on the target region although there is a sizable change in the amount of noise (which can be eliminated by shrink/expand noise cleaning). However, if one considers the sequence of best thresholds as determined by the border/edge match score (Table 3.10.1), there is a large shift (from gray level 27 to 17) even in this short sequence of frames. Thus no single threshold is appropriate for the whole sequence. Nonetheless, the previous threshold when used on the current frame is a fairly good choice. This suggests the following approach: In a single pass over the frame, segment the current frame using the best threshold(s) from the previous frame and simultaneously compute the best threshold(s) for this frame (to be applied to the next frame in sequence). The advantage of this scheme is that the frame is not stored, thereby realizing a considerable saving in chip size and complexity.



a.



b.

Figure 3.10.2a. Ten 64x64 windows from the sequential data base.

b. 5x5 median filtered windows of 128x128 originals, then sampled 2 to 1.

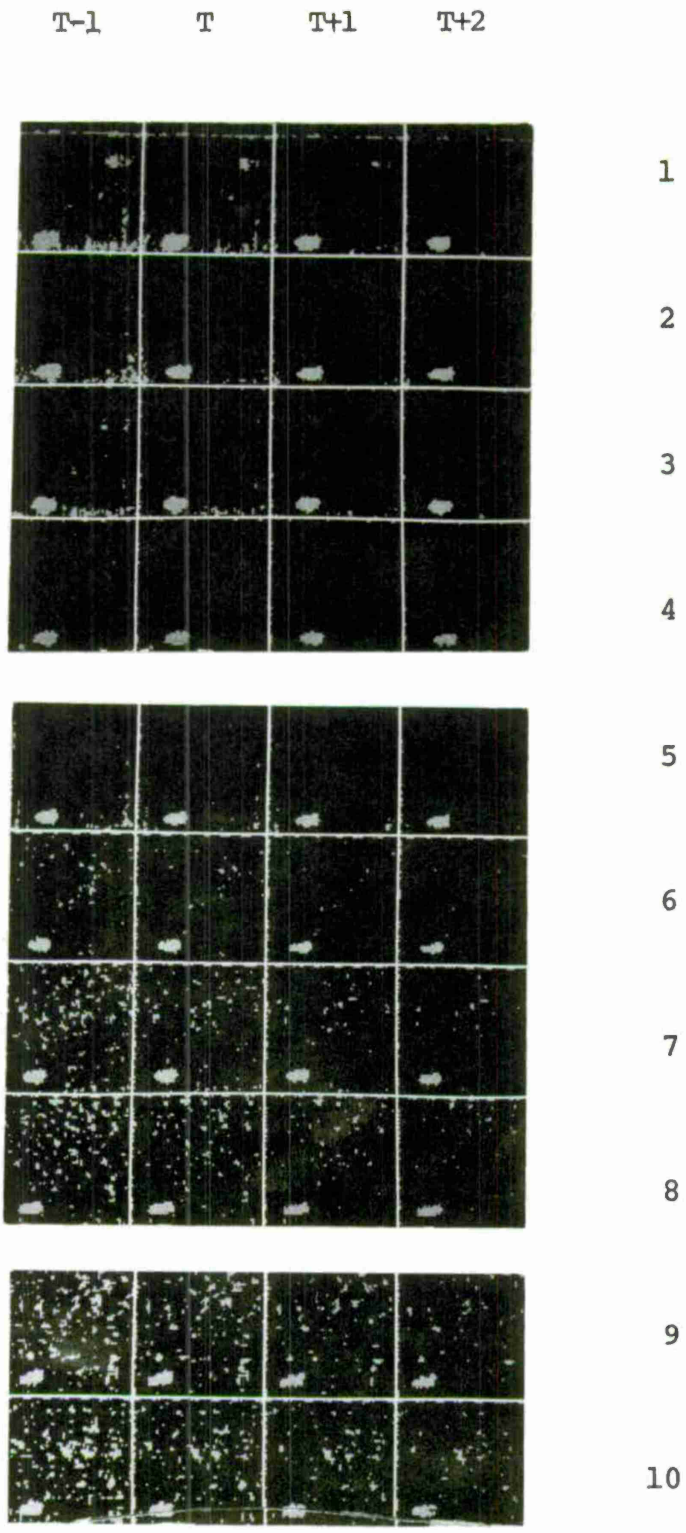


Figure 3.10.3. Effect of choosing lower or higher thresholds. The column labeled T shows the result of applying the chosen threshold to each window in the sequence. Columns T-1, T+1, T+2 show the results of using thresholds 1 lower, 1 higher, and 2 higher, respectively.

Threshold	Sequential Window #									
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
14	26	29	34	47	35	49	54	74	68	69
15	34	29	42	35	43	48	54	88	82	79
16	52	35	48	49	43	64	72	88	83	72
17	60	43	52	51	57	74	81	84	82	81*
18	53	51	72	58	72	72	90	90*	84*	69
19	54	53	76	67	70	76*	93*	87	71	65
20	59	60	85	75	72	59	89	66	75	65
21	67	67	87*	85	72	59	88	66		62
22	67	66	87*	100*	75*	50	80	63		50
23	67	70	87*	100*	68	42	80	63		
24	67	72	87*	97	68	38	80	63		
25	71	70	79	97	73	33	83	61		
26	76	76	81	85	69					
27	79*	79*	62	58	68					
28	76	77	64	54	68					
29	75	75	62	43	52					

Table 3.10.1. Percentage border/edge match as a function of threshold for the sequence data (maxima indicated with "*").

A somewhat different approach attempts to distribute N thresholds across the threshold range dynamically. Suppose the threshold range is x gray levels. It would take X/N frames to investigate each threshold in the range. However, as mentioned earlier, X/N is likely to be ≥ 3. Thus the entire gray level range capable of harboring targets can be sampled every 3 frames. At a projected processing rate of 3 frames per second, the range would be sampled once per second. A hybrid approach is also appropriate, devoting K of N thresholds to the most likely gray levels and letting N-K thresholds "rove" over the rest of the applicable gray-scale.

3.10.2 Region tracking

The Superslice algorithm builds a forest-like structure of regions from each frame. Within each structure, a sequence of nested regions which are roughly similar in size (but arising from different thresholds) constitutes a set of exemplars of a possible object. In addition, a certain number of accidents tend to be present. Regions of either type are called "candidate object regions". The frame to frame tracking process attempts to discover consistent temporal sequences of candidate object regions by selecting one exemplar per candidate object per frame, according to a dynamic programming model (see [28]).

Two evaluation functions, S and D, are used. The static evaluation function S(c) defines a figure of merit for each candidate object region c. The Superslice algorithm provides such a figure of merit based on contrast and well-definedness. The assumption is that the best exemplar for an object region is identified by having the greatest figure of merit. The dynamic evaluation function D(c,c') defines the similarity of one candidate object region (c) to another (c'). This is evaluated by considering the scaled differences between the feature vector of c and that of c'. If c is a perfect exemplar then S(c) = 0 and D(c,c) = 0.

Let $\{c_{ij};\ j = 1,\ldots,N_i\}$ be the set of candidate regions in the ith frame, i = 1,...,M. We define the dynamic programming problem as: find $\{c_{i\pi_i};\ i = 1,M\}$ such that

$T(c_{M\pi_M})$ is minimum over all selection functions, π.

The solution is achieved by the following:

Basis step: $T(c_{1j}) = S(c_{1j})$; $j = 1,\ldots,N_i$

Iterative step: $T(c_{i+1,j}) =$

$$S(c_{i+1,j}) + \min_{K=1}^{Ni} \{T(c_{ik}) + D(c_{ik},c_{i+1,j})\}$$

for $j = 1,\ldots,N_{i+1}$

The above procedure finds the minimum cost sequence of candidate object regions. Candidate regions which are accidental are unlikely to persist from frame to frame; thus their D terms are likely to be large, thereby increasing the total cost of any sequence which includes them. Note that there will be many sequences which are only slightly more costly than the minimum. These suboptimal sequences will be based on other exemplars for the same object. The optimal sequence is thus optimal for the particular formulations of S and D. Giving more weight to S and less to D will tend to select best exemplars; while the reverse weighting will tend to favor frame to frame consistency. A semantic model can provide guidance.

In general, the image sequence may contain more than one object. The scheme described above identifies the "best" object region sequence. In order to extract region sequences corresponding to other objects in the image sequence, we must delete all candidate object regions accounted for by

the optimal sequence. The inherent data structure specifies which regions are exemplars for each object. By deleting all candidate object regions in each frame which are similar to the selected region of the optimal sequence (i.e., contain it or are contained it it), we can set the stage for another application of dynamic programming. This process is repeated until only very poor (high cost) sequences are obtained. Presumably at this point all objects have been accounted for.

Occasionally, a deletion step may leave a particular frame empty of candidate object regions. This may occur for two reasons: All objects were accounted for by the last dynamic programming step, or the candidate region proposer failed to elicit an exemplar for an actual object. In the former case, the process will have terminated. The latter case can be handled by associating a fixed "empty frame" cost which is the price paid for skipping a frame. Of course, one can't know which case applied. The conservative approach is always to assume the second case and apply the empty frame cost. The termination criterion will then be based on a threshold for the total cost.

The problem of an object leaving the field of view can be handled in a different manner by flagging candidate object regions which lie on the border of the image. A partial sequence whose last element is flagged but which overall has low cost can be accepted as depicting an object which has moved off the image.

The dynamic programming algorithm described above has been implemented and tested on a sequence of ten windows of FLIR data containing a tank (Figure 3.10.4). These windows were already smoothed by a 3x3 median filter to provide better response to thresholding. The Superslice algorithm extracted a modest number of candidate object regions. Figure 3.10.5 displays these regions (although for nested sequences only the best static exemplar is displayed). The solution to the dynamic programming problem was computed and the exemplars which correspond to the solution are shown in Figure 3.10.6. There are of course many suboptimal solutions which are quite similar to this one. Their cost is not significantly greater than the minimal cost. When the indicated regions were deleted along with all other similar candidates, the only remaining regions corresponded to noise and any minimal cost path attempting to span several frames was substantially more costly than the optimal path or any of its similar suboptimal paths. It seems reasonable therefore to establish thresholds for static and dynamic cost in order to prune the search space.

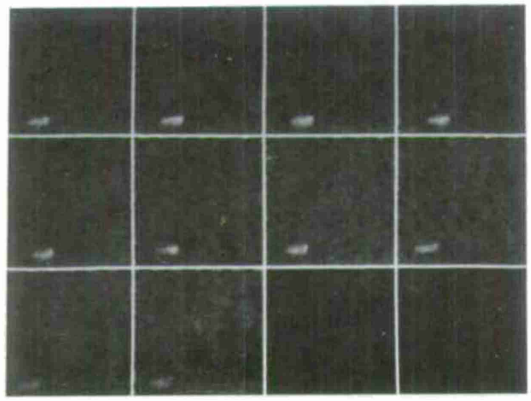


Figure 3.10.4. A sequence of 10 median filtered windows of a tank.

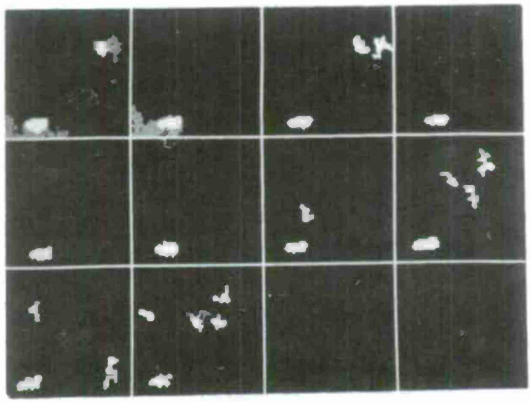


Figure 3.10.5. Output of the Superslice algorithm.

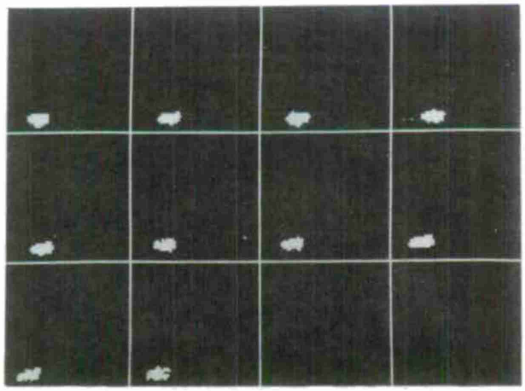


Figure 3.10.6. Optimal sequenced regions using dynamic programming.

3.11 Image processing software

3.11.1 Mini-XAP and Micro-XAP

Software engineering experience has demonstrated the many advantages of modularized design with clearly defined software interfaces. This approach was taken in the design of a picture processing software system named Mini-XAP. The system is intended for use as a flexible tool in general picture processing research efforts.

Mini-XAP is implemented on a PDP 11/45 minicomputer. Picture files can be stored locally on disk or 9-track tape. Pictures may be output to either a TV monitor or a precision CRT film recorder. Medium speed communication lines to UNIVAC 1106 and UNIVAC 1108 computers provide additional paths for picture data transfer and for program development activities.

Mini-XAP consists of a number of interface levels which mediate between the user, the operating system and the image processing subsystem. Briefly, it includes:

VOS, a multi-user virtual operating system developed at the University of Maryland;

DBM, a data base manager for image storage on disk and tape and for handling picture I/O;

ULI, the user language interface which allows communication between the DBM and LISP, the interactive high level programming language;

The algorithm skeleton, a program-like structure for common image transformations in which certain arguments and functions are left unspecified until the skeleton is executed;

Application packages, which offer the user access to predefined high-level image processing software.

A complete discussion of Mini-XAP is presented in [29].

Micro-XAP is a PDP-11/45 image access facility for Fortran users under the UNIX operating system. A more general facility, Virtual Arrays, has been developed for LISP users under UNIX. Documentation of these image handling software systems can be found in [30].

3.11.2 Viewmaster - a system for building image processing programs

Viewmaster is a generalized, concatenative local neighborhood image processing coordinator. The system is based on the premise that much image processing consists of sequences of transformations defined on local neighborhoods of each image point. Within a sequence, the output image of each transformation is fed into the next. A conventional approach to development of such programs is to create intermediate image products corresponding to the output of each transform. This tends to load the I/O channels and waste mass storage.

Our approach is to view local neighborhood operations as modules with similar input/output requirements. Saving of intermediate images is avoided by devoting extra core storage to buffer the needed rows of these "virtual" images. The generalized structure of the processing environment is as a set of cooperating processes with system-like requirements for resource allocation, race condition management, and memory utilization. The procedural control structure is represented by a directed acyclic graph. Pairs of adjacent processes communicate through data blocks describing the shared virtual image. Thus a virtual image is "produced" by a single process and "consumed" by one or more processes. This involves handling constraints which are not normally dealt with in buffer control systems.

A number of benefits derive from this approach.

- Image processing programs can be written without extensive knowledge of image I/O.
- Common image operations can be invoked by name (through a preprocessor) and concatenated.
- Program modules can be developed and tested independently and then integrated simply.
- Sharing of program modules is encouraged by the uniform operation environment.
- Available core storage is allocated and used efficiently. Mass storage serves as a back-up only as necessary.
- File and directory space for intermediate image products can be reduced or eliminated.

For a fuller description, see [31].

3.12 Recommended algorithm and flowchart

This subsection summarizes the various steps which make up the target detection algorithm described in Section 3.

Acquire the image from the sensor

Sample the image 2 to 1

Smooth the resultant image with a 5x5 median filter

Compute the edge map:

Apply a 4x4 difference-of-averages edge detection, selecting the maximum magnitude response over four directions

Thin the edge response using local directional non-maximum suppression

Compute a threshld range

Histogram the image

Identify the mode and the maximum significant gray level

Select K thresholds from the range

Evenly space K thresholds in the range

Build the containment forest of object region feature vectors as follows:

For each threshold (starting at the lowest):

Threshold the image

Extract feature vectors of connected components of above threshold points

Accumulate primitive features as each compon-

```
                ent is colored
                Select object region vectors using thresholds
                on size, edge/border cooccurrence and con-
                trast
                Make each vector point to its predecessor (if
                any) in the containment tree
        For each containment tree in the containment forest:
            For each exemplar vector:
                Classify the feature vector using the classi-
                fier
            If any feature vector in the containment tree has
            been classified as a target then signal a target
            cue for the corresponding region on the display
                Select the most "well-defined" exemplar in
                the containment tree (using the edge/border
                match criterion)
                Classify the target region according to the
                classification of this best exemplar (tank,
                truck, APC, unknown, small)
            Else classify the containment tree as noise
        IFEND
```

3.13 References

1. Algorithms and Hardware Technology for Image Recognition, First Quarterly Report, Computer Science Center, Univ. of Maryland, College Park, MD, July 1976.

2. Algorithms and Hardware Technology for Image Recognition, First Semiannual Report, Computer Science Center, Univ. of Maryland, College Park, MD, October 1976.

3. Algorithms and Hardware Technology for Image Recognition, Second Semiannual Report, Computer Science Center, Univ. of Maryland, College Park, MD, April 1977.

4. Algorithms and Hardware Technology for Image Recognition, Third Semiannual Report, Computer Science Center, Univ. of Maryland, College Park, MD, October 1977.

5. Panda, D. P., "Segmentation of FLIR Images by Pixel Classification", University of Maryland, Computer Science TR-508, Feb. 1977.

6. Panda, D. P., "Statistical Properties of Thresholded images", University of Maryland, Computer Science TR-559, July 1977.

7. Panda, D. P., "Statistical Analysis of Some Edge Operators", University of Maryland, Computer Science TR-558, July 1977.

8. Hueckel, M., "A Local Visual Operator Which Recognizes Edges and Lines", JACM, Vol. 20, 1973, pp. 634-647. [Erratum: JACM, Vol. 21, 1974, p. 350.]

9. Hueckel, M, "An Operator Which Locates Edges in Digitized Pictures", JACM, Vol. 18, 1971, pp. 113-125.

10. Hummel, R. A., "Edge Detection Using Basis Functions", University of Maryland, Computer Science TR-569, August 1977.

11. Mero, L., Vassy, Z., "A Simplified and Fast Version of the Hueckel Operator for Finding Optimal Edges in Pictures", Proc. 4th Intl. Conf. on Artif. Intelligence, Tbilisi, USSR, Sept. 1975, pp. 650-655.

12. Shaw, G. B., "Local and Regional Edge Detectors: Some Comparisons", University of Maryland, Computer Science TR-614, December 1977.

13. Peleg, S., "Iterative Histogram Modification, 2", University of Maryland, Computer Science TR-606, November 1977.

14. Davis, L. S., "A Survey of Edge Detection Techniques", Computer Graphics and Image Processing, Vol. 4, 1975, pp. 248-270.

15. Weszka, J. S., Rosenfeld, A., "Threshold Selection Using Weighted Histograms", University of Maryland, Computer Science TR-567, August 1977.

16. Milgram, D. L., Herman, M., "Clustering Edge Values for Threshold Selection", University of Maryland, Computer Science TR-617, December 1977.

17. Nakagawa, Y., Rosenfeld, A., "Some Experiments in Variable Thresholding", University of Maryland, Computer Science TR-626, January 1978.

18. Chow, C. K., Kaneko, T., "Automatic Boundary Detection of the Left Ventricle From Cineangiograms", Comput. Biomed. Res. 5, 1972, pp. 388-410.

19. Nakagawa, Y., Rosenfeld, A., "A Note on the Use of Local MIN and MAX Operations in Digital Picture Processing", University of Maryland, Computer Science TR-590, October 1977.

20. Milgram, D. L., "Constructing Trees for Region Description", University of Maryland, Computer Science TR-541, May 1977.

21. Rosenfeld, A., "Fuzzy Digital Topology", University of Maryland, Computer Science TR-573, September 1977.

22. Dyer, C. R., Rosenfeld, A., "Thinning Algorithms for Grayscale Pictures", University of Maryland, Computer Science TR-610, November 1977.

23. Ohlander, R., "Analysis of Natural Scenes", Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, PA, December 1976.

24. Milgram, D. L., Kahl, D. J., "Recursive Region Extraction", University of Maryland, Computer Science TR-620, December 1977.

25. Stockman, G. C., "Maryland Interactive Pattern Analysis and Classification System, Part I: Concepts", Dept. of Computer Science, University of Maryland TR-408, College Park, MD, September 1975.

26. Wertheimer, M., "Principles of Perceptual Organization", in Readings in Perception, D. C. Beardlee and M. Wertheimer (eds.), p. 122, Van Nostrand-Reinhold, Princeton, NJ, 1958.

27. Abend, K., "Compound Decision Procedures for Unknown Distributions and for Dependent States of Nature", Pattern Recognition, L. Kanal, Ed., Washington, DC, 1968, pp. 207-249.

28. Milgram, D. L., "Region Tracking Using Dynamic Programming", University of Maryland, Computer Science TR-539, May 1977.

29. Luczak, E. C., Hayes, K. C., "Mini-XAP Image Processing System", University of Maryland, Computer Science TR-601, November 1977.

30. Hayes, K. C., Herman, M., Smith, R., "PDP-11 Image Processing Software", University of Maryland, Computer Science TR-612, December 1977.

31. Dyer, C. R., Milgram, D. L., "Viewmaster - A System for Building Image Processing Programs", Proc. of 8th Annual Symposium on Automatic Imagery Pattern Recognition, pp. 170-179, April 1978.

4. HARDWARE TECHNOLOGY

This section of the final report was prepared by the Westinghouse Systems Development Division for the Computer Science Center, University of Maryland. It contains a review and summary of a 21-month effort to generate and implement automatic target cueing algorithms, involving three phases, as follows:

Phase I: Task and Technology Review (3 months)
Phase II: Algorithm Selection and Test (9 months)
Phase III: Hardware Development (9 months)

All phases of the program were completed during the prescribed period, including the construction and demonstration of an important recognition function using charge-coupled devices.

The success of the program is due, in large part, to the close coordination between members of the government-university-industry team. The principle team members from the government were Mr. John Dehne and Dr. George Jones of NVL. For the University of Maryland, principle members were Profs. Azriel Rosenfelt and David Milgram. The Westinghouse team included Dr. Glenn Tisdale, Program Manager, Mr. Thomas Willett, Project Engineer, Dr. Nathan Bluzer, and Dr. Gerald Borsuk.

This section of the report was prepared by Thomas Willett and Glenn Tisdale.

4.1 Introduction

We will begin this section by reviewing the nature of and the need for automatic target cueing. This will lead to a specification of system design goals in Section 4.2. Algorithm design considerations will be examined in Section 4.3, including both system data processing requirements and their implementation in suitable hardware. The process of hardware fabrication will be covered in Section 4.4. Conclusions and recommendations are offered in Section 4.5.

Definition of Automatic Target Cueing

Before discussing the design goals and hardware implementation for an automatic target cueing system, it is useful to examine its character and purpose.

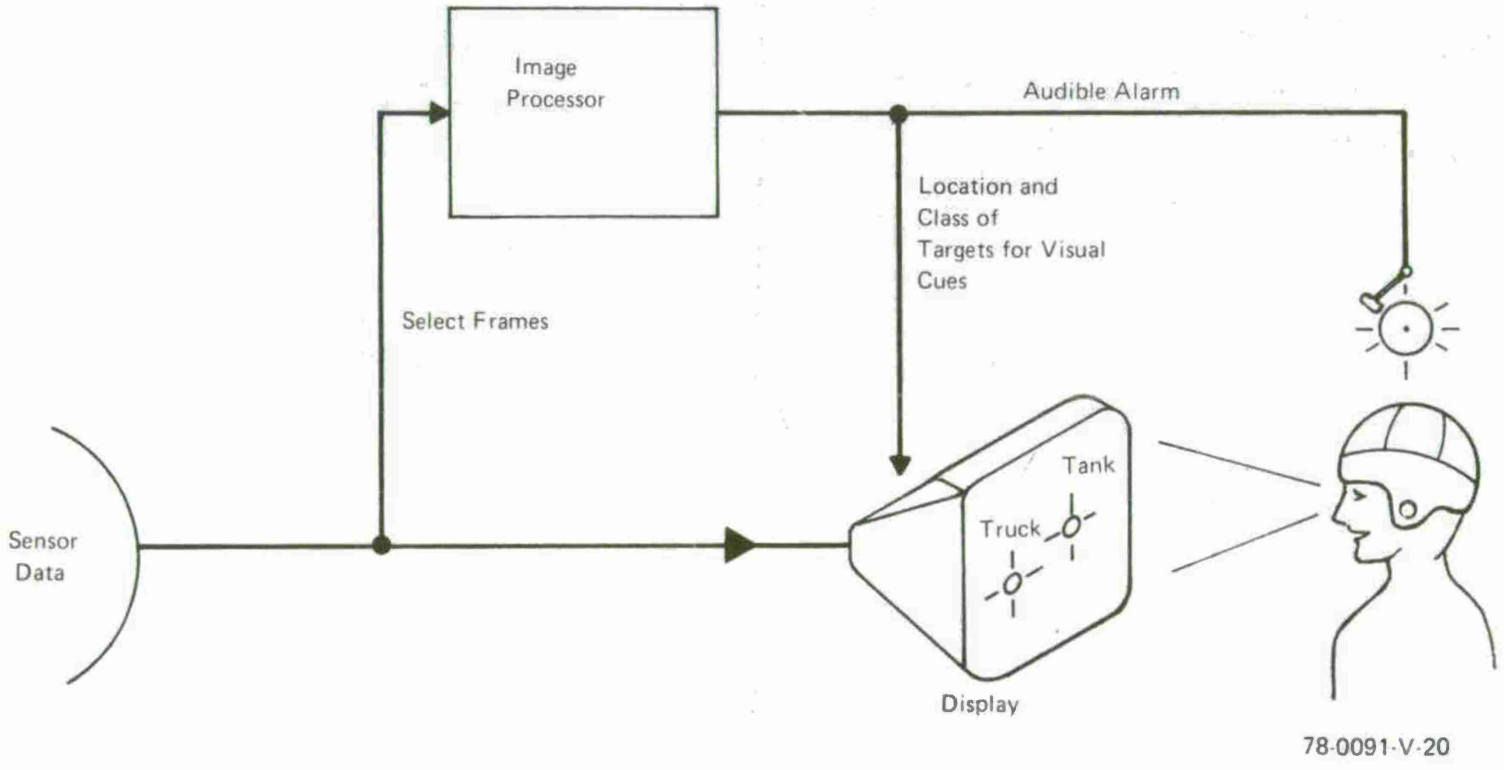


FIGURE 4-1: The Automatic Cueing Function

As the tools used in reconnaissance and target acquisition operations continue to improve, it is becoming increasingly clear that man himself is a performance bottleneck. Sensors provide greater resolution, dynamic range, and speed; and the techniques that make effective use of sensor information, such as communication links and weapon delivery systems, are also being advanced. However, acquisition of target material largely depends on the ability of a human operation to search sensor imagery in real time. If he misses targets, or identifies them incorrectly, mission performance is degraded.

One way to improve this situation is to provide the human with cues to locate, and possibly identify, targets on his displays. Such cues might originate with a priori information, or from the detection of peculiar target conditions such as motion, hot spots, or electromagnetic radiation. However, a more general approach is to apply automatic recognition techniques to the imagery. As shown in Figure 4-1, the image processor serves as an information filter on the image, alerting the human to the presence of potential targets, possibly by audible signals initially, and then by providing visual cues or overlays on his display. In this process, the final verification of target identity is reserved for the human operator. The image processor serves to assist him.

Automatic cueing can be carried out either in airborne or ground locations. In the airborne situation, the operator views a CRT-type display for acquisition of targets on a real-time basis. His determination may result in action in a matter of seconds, either offensive or defensive. On the ground, interpretation may be required in real-time, or on a more relaxed basis. In the proposed operation of remotely piloted vehicles (RPV's), for example, a video link may be used to obtain a CRT presentation at a ground station of the output of a sensor located on the vehicle. The problems for the operator are somewhat similar to the airborne situation; however, his appraisal of the sensor image is entirely limited to the CRT output. He can't look at the target area directly. On the other hand, he is not distracted by problems such as vehicle operation and personal security.

In a more relaxed mode, cueing of wide area high resolution reconnaissance imagery might be used to greatly reduce the time required in scanning imagery. In this case, the results of the processing is desired in a matter of minutes, so as to respond to the motion of mobile targets.

4.2 System Design Goals

Key considerations in the design of an automatic target cueing system are its performance, physical characteristics, and allowable cost. We will deal with each of these considerations in this paragraph. However, a quantitative determination of design parameters will depend on the manner in which the mission is implemented. Such implementation will be discussed first.

As explained in Section 4.1, the target cueing function might be performed aboard a vehicle, or at a ground station if imagery is relayed for analysis. In either case, the performance goals will tend to be comparable. As regards physical characteristics and cost, however, the vehicle location will demand much tighter restrictions. Our discussion will proceed on the basis of the vehicular application. Both helicopters and high-speed aircraft are airborne candidates. The RPV image, on the other hand, will be analyzed at a ground station; therefore, the physical limitations within the RPV are not a problem. As the state of the art in automatic target recognition develops, and high levels of performance are attained, it is anticipated that the human observer will eventually be eliminated in some applications. For example, recognition equipment might be placed aboard a missile for unaided terminal guidance. The requirement for high performance, small size and weight, low power consumption, and low cost will all apply in this case.

Performance Goals

Key performance parameters are the detection and recognition rates for targets of interest, the false alarm rate, and the speed of operation of the cueing system. Detection occurs when a target of any kind is indicated by the cueing system, while recognition occurs when the correct target class is selected from among several possible classes. Detection and recognition performance is expressed as a percentage of the targets which are actually available. A false alarm occurs when a target is indicated even though none is present. The false alarm rate is expressed on the basis of a unit of elapsed time or area of coverage. The required speed of operation is determined by the time available to the operator to make decisions, the search area to be covered by the sensor, and the sensor frame rate. It depends heavily on human factors considerations, such as decision times and reaction times, and the choice of prioritization ground-rules.

In the discussion in the first quarterly report,[1] it was noted that the cueing system must significantly improve the search capability of the unaided human interpreter if it is to be cost effective. The problem is to locate targets which occupy a tiny area in the sensor field of view, and are embedded in background clutter and noise. The performance of the interpreter is limited by visual scanning rates, eye fatigue, and conflicting duties. References were cited which tend to indicate rather low target acquisition rates (such as 33 percent detection and 25 percent recognition) under realistic conditions. Although no data are available which provide a direct comparison of operator performance with and without cueing, there is growing evidence that cueing systems could more than double this performance.[2,3] If this evidence is substantiated, then the outlook for a cost-effective cueing system appears very promising. However, a final determination depends on combining specific system and mission parameters.

We are particularly concerned with the airborne mission, where space and weight available for the equipment are at a premium. Mr. Donald Looft, Deputy Director of DARPA, has described airborne scenarios for helicopters and high-speed aircraft as follows:[4]

> "Now visualize a likely scenario for the one-man pilot observer. He must direct the sensor and visually scan his display for targets which most assuredly are concealed by terrain or background clutter. He must fly low and as fast as possibly to minimize exposure to radar controlled weapons, say in the helicopter case, speeds of 60 to 80 knots at altitudes of 100 to 200 feet are typical. In the case of attack aircraft, speeds of 400 to 500 knots at altitudes of two to three kms. (6560 - 9840 feet) are normal flight profiles, though the latter can also be "down on the deck".

[1] Algorithms and Hardware Technology for Image Recognition, Westinghouse Quarterly Report on Contract DAAAG53-76-C-0138, 1 May through 31 July 1976, page 2-2.

[2] Tisdale, G.E., A Digital Image Processor for Automatic Target Cueing, Navigation, and Change Detection, Proceedings of the SPIE Symposium on Tactical Reconnaissance, Reston, Virginia, April 20, 1977.

[3] Algorithms and Hardware Technology for Image Recognition, Quarterly Management Report by Univ. of Maryland, 1 August - 31 October, 1977, Par. 2.3.

[4] Looft, D.J., Image Understanding -- A Perspective of DOD Needs, banquet speech presented at the Sixth Annual AIPR Symposium, College Park, Maryland, June 1, 1976.

"At the same time, the man or the crew is performing these flight maneuvers which are "hairy" in broad daylight, let alone darkness, he must monitor his aircraft instruments, he must be wary of terrain obstacles, and he may well be under enemy fire. Ideally, the pilot wants to acquire and engage targets on a single pass. If he must take several passes, his survivability is greatly reduced. If he has been directed to a probable target area and is able to take advantage of terrain masks he must "pop up", make his observations, acquire and engage targets in times like 15 or 20 seconds. He would always like to acquire and engage enemy targets at sufficient standoff distance to be out of range of radar controlled enemy air defense weapons. This means he is usually at or near the threshold recognition range of the sensor."

A detailed examination of the trade-offs between the required cuer processing rate and the allowable false alarm rates was presented recently by Dehne et al. of NVL[5]. For a set of mission parameters which relate primarily to the helicopter scenario, it was found that for processing rates between 3 and 10 frames per second, from 0.5 to 1.8 false alarms could be accommodated per frame. These results assumed that 20 seconds were available to cover a large search window, resulting in about 0.7 seconds to handle each frame. This figure includes the frame processing time, the time to slew the sensor, the operation decision time per false alarm (0.2 seconds) and his reaction time to advance to the next frame (0.2 seconds). The report considers sequential as well as combined processing and slew, with the former preferred.

A separate consideration with the above processing rates is that the cueing symbols superimposed on the display must appear in the correct location even after the processing delay. With a frame rate of 30 per second, the delay covers 3 to 10 frames (0.1 to 0.3 seconds). Misregistration of target and symbol could be caused by target motion relative to the terrain, or by the changes in the field of view due to aircraft motion. Broadside motion of the target is generally the worst case. Suppose the cueing window subtends twice the extent of the target on the display in both the horizontal and vertical dimensions, and is located at the point of the target center in the processed frame. It can move by half its dimension in any direction and still be contained within the cueing window. For a vehicle 20 feet in length which subtends 20 pixels in the display,

[5]Dehne, J.S., Van Atta, P. and Raimondi, P., Specifying Image Processing Systems for Thermal Imagers, paper presented at the Seventh Annual Symposium of the EIA-AIPR Committee, College Park, Md. 22-24 May 1977.

motion of 10 feet must be accommodated over a worst case period of 0.3 seconds, with a corresponding allowable broadside speed of 30 feet per second (43 mph). This result is independent of range if the window is proportional to target size.

The report also considers the use of a wide sensor field of view for cueing, followed by operator confirmation with the narrow field of view. It is assumed in this case that the capability of the cuer for recognition exceeds that of the operator by an amount sufficient to compensate for the increased field of view. Under these conditions, one false alarm per frame could be accommodated with a processing rate of 0.54 frames per second (about a 10:1 reduction over the previous case). However, at the present state of the art, this improved cuer performance relative to the operator has not been demonstrated. In that regard, it is noted that because of the eye integration time of 0.2 seconds, the operator gains a signal-to-noise improvement over the cuer of, perhaps, 2.5.

A final approach considered a sensor with an expanded, high resolution scan area equivalent to the target search window. Due to display limitations, the operator sees either a low resolution version of the entire window, or a small segment containing potential targets as selected by the cuer. The cuer processing rate is not greatly affected by this mechanization.

For the assumed frame size of 375 by 500 pixels, the processing rates of 3 to 10 frames per second correspond to data handling rates of 0.6 to 1.9 megapixels per second.

The foregoing discussion was addressed to the helicopter scenario. For the high-speed aircraft, the available search time will tend to be lower, but the required search window will probably also be reduced. The reduced search window can be achieved by reliance on navigation aids for the acquisition of predesignated targets. At the high speeds and possible low altitudes, it appears that the single-seat operator, because of the burden of aircraft navigation, will be assisted significantly by the cueing system. Frame rates which are comparable to his reaction time, or somewhat lower, in the vicinity of one per frame should be tolerable from the point of view of overall processing time. However, from the point of view of increased detection and recognition rates, as well as reduced false alarm rates, it appears useful to consider the

integration of results from successive frames. The assignment of a priority weighting to target cues will improve the probability that important targets will be considered when a number of opportunities occur.

Physical Characteristics

The significant physical characteristics of the cueing system for aircraft or missile use include size, weight and power consumption.

The increased availability of general-purpose MSI circuits has made it possible to offer existing cuer algorithms, using conventional packaging techniques, in packages which should be acceptable for aircraft use. A total system, excluding displays, might be expected to occupy a volume of 0.5 to 1.0 cubic feet, and to weigh 15 to 30 lbs., including power supply. Power in the neighborhood of 200 to 300 watts will be required.

For missile applications, conventional packaging can be improved upon by use of flat packs, or bare chip packaging, and by the introduction of some specially designed chips.

One thrust of the present Westinghouse program, however, has been to determine the necessary area of silicon substrate required to provide cuer functions. As will be described in Section 4.4, the fabrication of special CCD LSI circuits appears feasible, and would reduce the cuer to an overall chip area of a few square inches. On this basis, introduction of cueing functions into an artillery shell, for "fire-and-target" performance, becomes feasible.

Allowable Cost

In Section 2.4 of the first quarterly report on this program, an attempt was made to relate the allowable cuer cost to its mission value. The argument held that if the operator's performance was significantly improved with the use of cueing, then the value of cueing becomes a significant fraction of the overall cost of flying a series of missions (a very large figure). Unfortunately, the improvements associated with cueing have not yet been measured, and the assignment of an improvement factor will no doubt be elusive. Furthermore, an upper cost limit, which is much lower, can realistically be anticipated for any new addition to a vehicle complement.

Since the cueing system is a digital processor, its production cost using conventional packaging techniques can be estimated reasonably well. For helicopter or aircraft use, a figure of $20K to $50K per unit is suggested. Reductions in size by increased use of LSI will tend to reduce the recurring cost for each unit, but at the expense of a significant nonrecurring cost for initial development.

Implementation of a complete cueing system, using CCD circuits on a small number of silicon chips, takes this sequence one step further. The final unit recurring cost, in production, might range from $1K to $5K, including test, but the development program would be a multimillion dollar affair. Before such an investment could be considered, a number of hurdles would have to be overcome. First, the satisfactory operation of the system would have to be established. Next, an attempt should be made to compare the performance of competitive approaches, since only one design can probably be initiated. Finally, a variety of applications should be considered, so that the development cost can be divided as much as possible.

A practical approach to this dilemma, which has been initiated in the present program, consists of the selection and implementation of key circuit functions in CCD form. These circuits can hopefully be used in hybrid arrangements to reduce the size, weight, and power of early cuer designs. With the growing availability of new chips, these values will continually decrease. At the same time, the solution of a variety of application problems will be possible from the library of available chip designs.

4.3 Algorithms, and Hardware Implementation

This section describes a preferred set of algorithms developed by Maryland which tentatively comprises the first portion of a cueing system. A system flowchart is shown in Figure 4.3-1. A description of data flow and storage requirements is included.

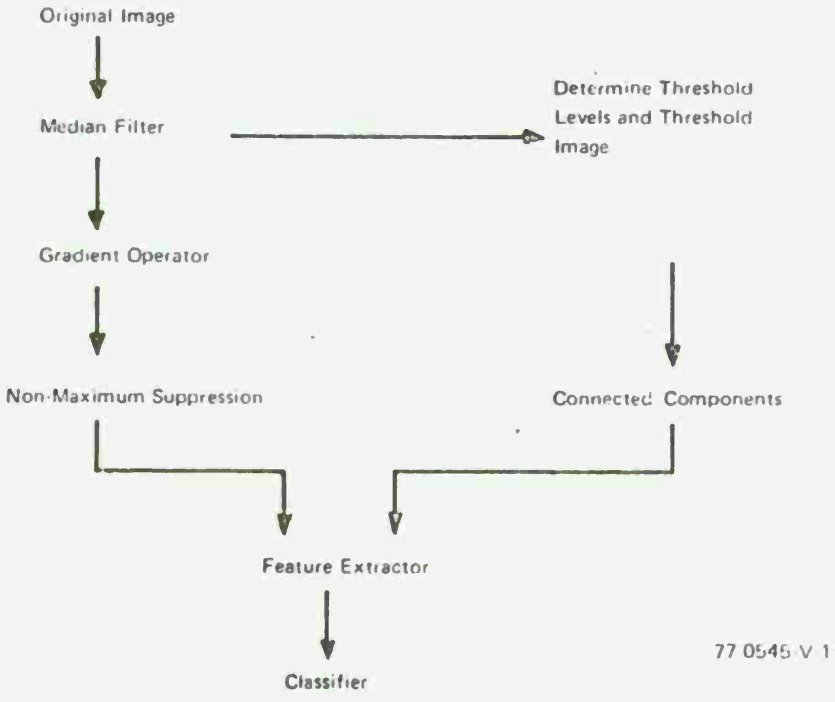


Figure 4.3-1. System Flowchart

In general, the Median Filter acts to suppress noise. The Gradient Operator extracts edges which are then thinned by the Non-Maximum Suppression Algorithm. At the same time a set of gray levels is determined and the filtered image is thresholded at each gray level. A Connected Components Algorithm partitions each of the thresholded images into potential object regions. The Super Slice Algorithm correlates perimeter points formed independently by the Non-Maximum Suppression and Connected Components Algorithms and a score is obtained for each gray scale threshold. These scores and several other algorithms form a set of Classification Logic. A short description of each algorithm follows.

4.3.1 Gradient Operator

The Gradient Operator is an edge detector which is defined as GRAD OP = max $\{|A-B|, |C-D|\}$ where A, B, C and D each represent the sums of overlapping regions of 4 X 4 pixels each, as seen in Figure 4.3-2.

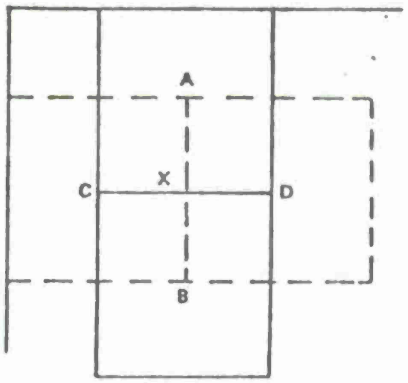


Figure 4.3-2. Gradient Operator

The value of GRAD OP is placed in the pixel location marked "x" which is one pixel to the left and above the center of the entire region. The key arithmetic operation in GRAD OP is the formation of the difference

$$A-B = \begin{cases} 0 & \text{if } |A| < |B| \\ A-B & |A| > |B| \end{cases} \qquad (4.3\text{-}1)$$

which is realized on a silicon substrate with the configuration shown in Figure 4.3-3. D_{in} is a diffusion diode through which charge is injected into the chip; A and B are gates whose potentials are controlled by voltages representing the sums A and B, respectively. These gates will form a trap to retain some of the injected charge. The trapped charge is equal to A-B and is removed by the transfer gate.

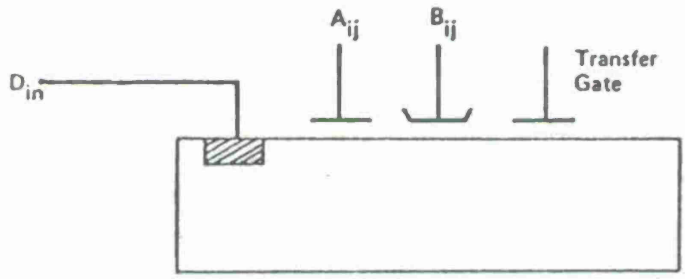


Figure 4.3-3. Subtraction Module

Let us consider a physical analogy in which Din, A, B, and TG are a set of steps, the height of each step represents the potential, and the charge is represented as water.

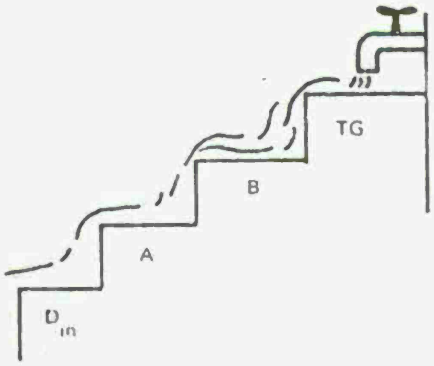


Figure 4.3-4a. No Water Trapped,

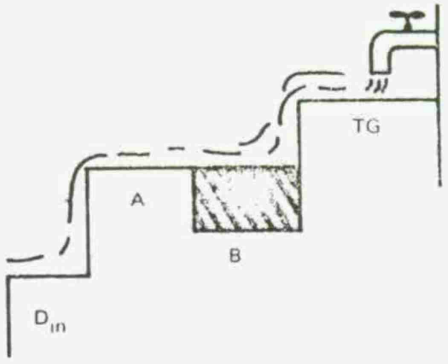


Figure 4.3-4b. Water Trapped

When the height (potential) of step (gate) A is raised above that of step (gate) B, water (charge) is trapped above B in an amount corresponding to A-B. Figure A 4.3-4 a, b represents both conditions of Equation (4.3-1).

$$\text{The algorithm } B-A = \begin{cases} 0 \text{ if } |B| < |A| \\ B-A \text{ if } |B| > |A| \end{cases}$$

requires another silicon substrate in which the gate positions of A and B are reversed. The block diagram of the absolute difference operator $|A-B|$ is shown in Figure 4.3-5.

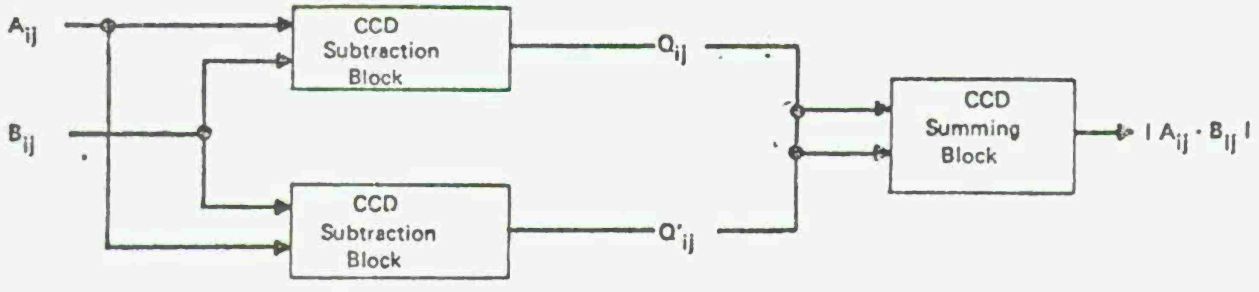


Figure 4.3-5. Absolute Value Circuit

If $|A| < |B|$, then $B_{out} = 0$, $A_{out} = B-A$ and $A_{out} + B_{out} = B-A$. Similarly, if $|A| > |B|$, then $B_{out} = A-B$, $A_{out} = 0$ and $A_{out} + B_{out} = A-B$. These two statements form an absolute value operator. A similar operation can be formed for $|C-D|$, but GRAD OP really compares $|A-B|$ and $|C-D|$ and chooses the maximum since it is defined as max $\{|A-B|, |C-D|\}$. Figure 4.3-6 shows the GRAD OP block diagram, and Figure 4.3-5 shows the outpu s at various locations.

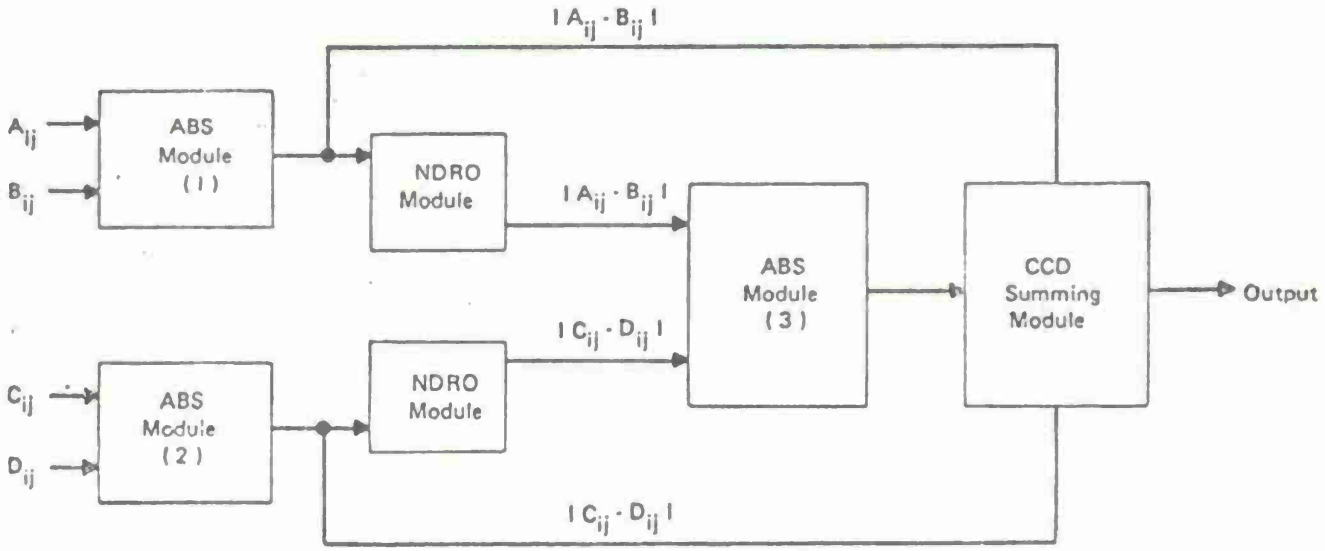


Figure 4.3-6. CCD Gradient Operator

Condition	$\|A-B\| < \|C-D\|$	$\|C-D\| < \|A-B\|$
ABS Module (3) Output	$\|C-D\| - \|A-B\|$	$\|A-B\| - \|C-D\|$
Summing CCD input	$\|C-D\| + \|A-B\|$	$\|C-D\| - \|A-B\|$
Outputs From Summing CCD	$2\|C-D\|$	$2\|A-B\|$

Figure 4.3-7. Output of GRAD OP at Various Locations

4.3.2 Median Filter

The Median Filter acts to extract the median value from a 5 x 5 pixel window and place that value in the center pixel location; the Filter can be considered as a smoothing operator. The Filter quantizes each of the 25 analog signals into a number of discrete units and then sorts the 25 quantized signals by arranging them in a descending order of magnitude.

a. Charge Quantizer

The silicon substrate forming the Quantizer is shown in Figure 4.3-8.

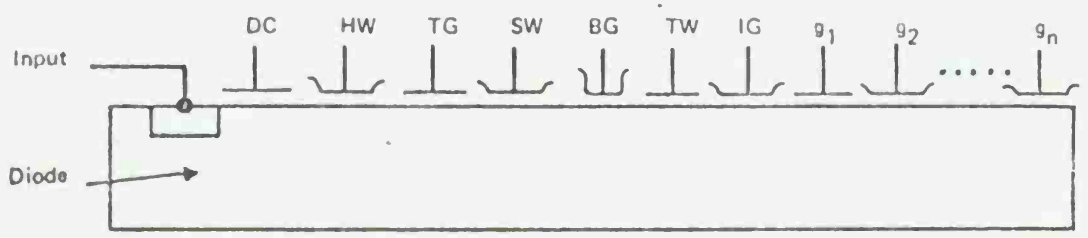
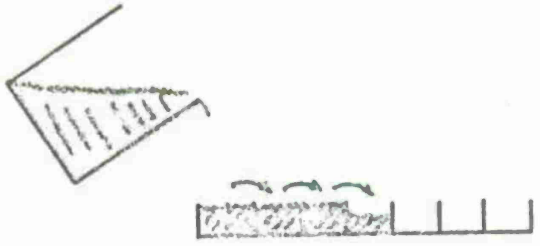


Figure 4.3-8. Charge Quantizer. Symbols are Identified in the Text

D_{in} is the diffusion diode through which charge is injected into the chip and into the holding well, HW. DC blocks the charge from leaving via D_{in}. An amount of charge, Q, proportional to S, the signal voltage, is removed from HW via the transfer gate, TG, and placed in the signal well, SW. Via the blocking gate, BG, and the thimble well, TW, a discrete amount of charge, q, is removed and placed in well g_1. Another quantum q is removed from SW, and placed in g_1 while the first charge has been shifted to g_2. This process is repeated until SW is empty and all the charge has been placed in a number of discrete wells g_1, g_2 ... g_n.



77-0545-V-16

Figure 4.3-9 Flow Analogy to Charge Quantizer

proportional to the signal voltage S,being poured into a tray of quantized bins. When a bin is filled with water the water flows over the top into the next bin. The volume of water is divided between a number of discrete bins.

b. Sorter

Recall that the contents of wells g_1, g_2, ... g_n of the quantizer (Figure 4.3-8)each contain, at most, an amount of charge q. The contents of each well is parallel shifted into its corresponding channel of the sorter (Figure 4.3-10) so that if there were q charge in g_3, there is now q charge in I_3 and so on. Forming traps as described in Section 4.3.1 with wells P_{g1}, P_{g2}, and P_{g3},the charge in each channel is shifted into the large holding well (LHW). The large holding well is partitioned into N channels also. Consider a numerical example; a sequence of numbers 4, 7, 5 is quantized at q = 1 so that 4, 7, and 5 levels respectively represent each number. Then Figure 4.3-11 shows the sequence as it goes through the quantizer; the g_1, g_2, ... g_n wells; the I_1, I_2, ... I_n

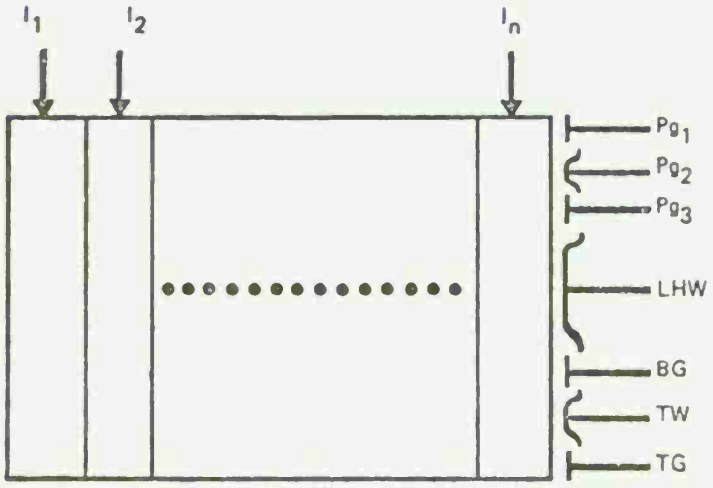


Figure 4.3-10 CCD Sorter

channels; and the large holding well. It also shows the removal sequence from the large holding well and the remainder at each stage. The numbers are removed in order of decreasing magnitude 7, 5, 4 which shows the numbers have been sorted by magnitude.

4.3.3 Non-Maximum Suppression

The Gradient Operator extracts edges in either the horizontal or vertical direction; the Non-Maximum Suppression Algorithm then looks in a direction perpendicular to the edge for a larger gradient. If a larger value cannot be found, the edge under consideration is retained; the edge is removed if a larger value is found. The Algorithm is shown in Figure 4.3-12.

	1	2	3	4	5	6	7	8	9 ..... n
g_i Wells									
5	q	q	q	q	q				
7	q	q	q	q	q	q	q		
4	q	q	q	q					
I_i Channels									
5	q	q	q	q	q				
7	q	q	q	q	q	q	q		
4	q	q	q	q					
Large Holding Well									
4	q	q	q	q					
4,7	2q	2q	2q	2q	q	q	q		
4,7,5	3q	3q	3q	3q	2q	q	q		
First Removal									
	q	q	q	q	q	q	q		
Remainder									
	2q	2q	2q	2q	q				
Second Removal									
	q	q	q	q	q				
Remainder									
	q	q	q	q					
Third Removal									
	q	q	q	q					
Remainder									

Figure 4.3-11 Sorting Sequence

```
x x x
x x x
  x
  y      →
  x
x x x
x x x
```

If any x>y, y = 0

Otherwise retain y

x, y are gradient values

Figure 4.3-12. Arrangement of Comparison for Non-Maximum Suppression

The gradient under consideration is a vertical one and the area examined for larger gradients is in the vertical direction.

Embodiment of the Non-Maximum Suppression Algorithm (NMS) requires several operations with CCD structures. A key part of NMS is extracting the largest x_m gradient value in the neighborhood surrounding y; x_m is then compared to the gradient value y_g representing the y^{th} pixel. Sorting the x values to obtain x_m can be accomplished by the sorting operation described earlier. Comparing x_m and y_g can be done by the subtraction module also described before. Then the block diagram appears in Figure 4.3-13.

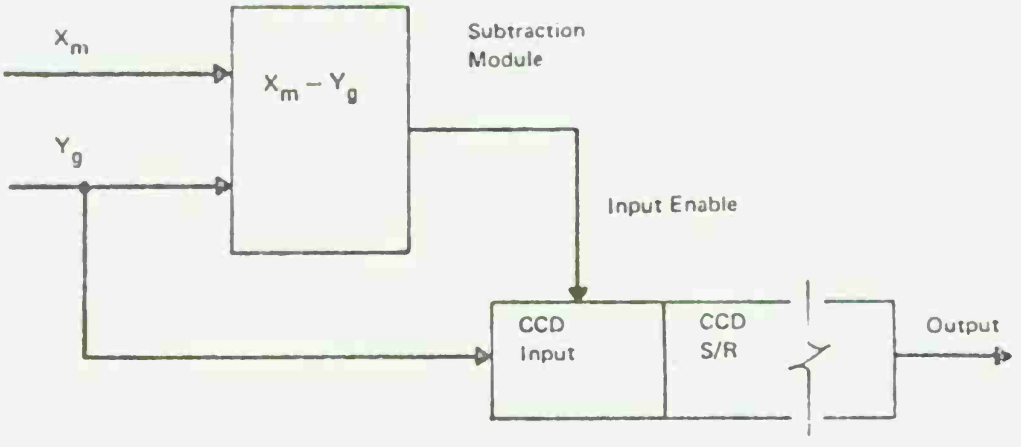


Figure 4.13. NMS Block Diagram

This time the subtraction module outputs an enable signal to the CCD shift register instead of the actual difference. A blocking gate shown in Figure 4.3.-14 is used to block (enable) y_g from entering the register.

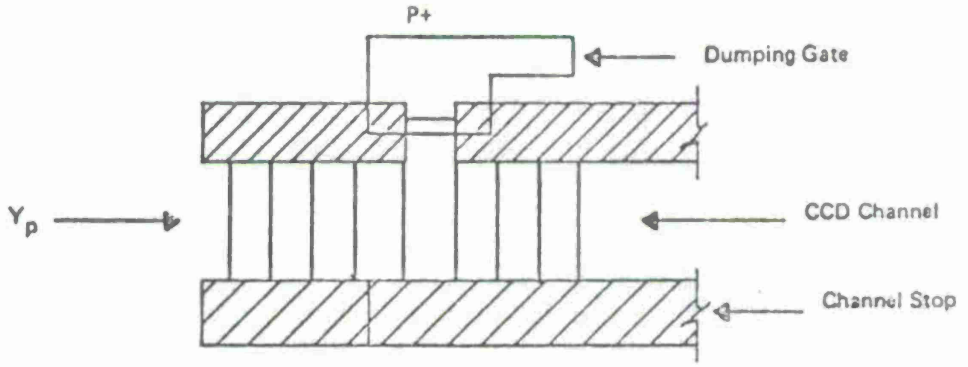


Figure 4.3.-14 Blocking Gate

4.3.4 Threshold Determination

The philosophy here is not to attempt to find a single threshold but rather to use a set of thresholds which span the range of gray scale. For the NVL FLIR data, fifteen (15) gray scale levels represented the gray scale ranges and selecting a threshold every three gray scale levels was deemed to give satisfactory target detections by the University of Maryland.

To implement this type of algorithm would require a sorter (as described in section 4.3.1.2 which arranges the gray levels in descending order for the image. The first number (the largest) leaving the sorter corresponds to the first threshold and sets a counter to 1. The second exiting number is compared with the first and the counter updates to 2 if the second number is different. If it is the same, the counter remains at 1. In general, each number is compared with the previous one to determine if the counter should be updated. When the counter gets to 3, the second threshold level is determined. In this manner every third gray scale level is selected and used as a threshold. A block diagram of the implementation is shown in Figure 4.3-15.

4.3.5 Connected Components

The purpose of the algorithm is to segment the image data stream into smaller domains. Each small domain includes a single object in the image plane. This algorithm distinguishes between objects and isolates regions so that statistics for Classification Logic can be obtained.

Assume that the original image has been thresholded and the result is in binary form with gray levels exceeding g_1 shown as 1's in Figure 4.3-16.

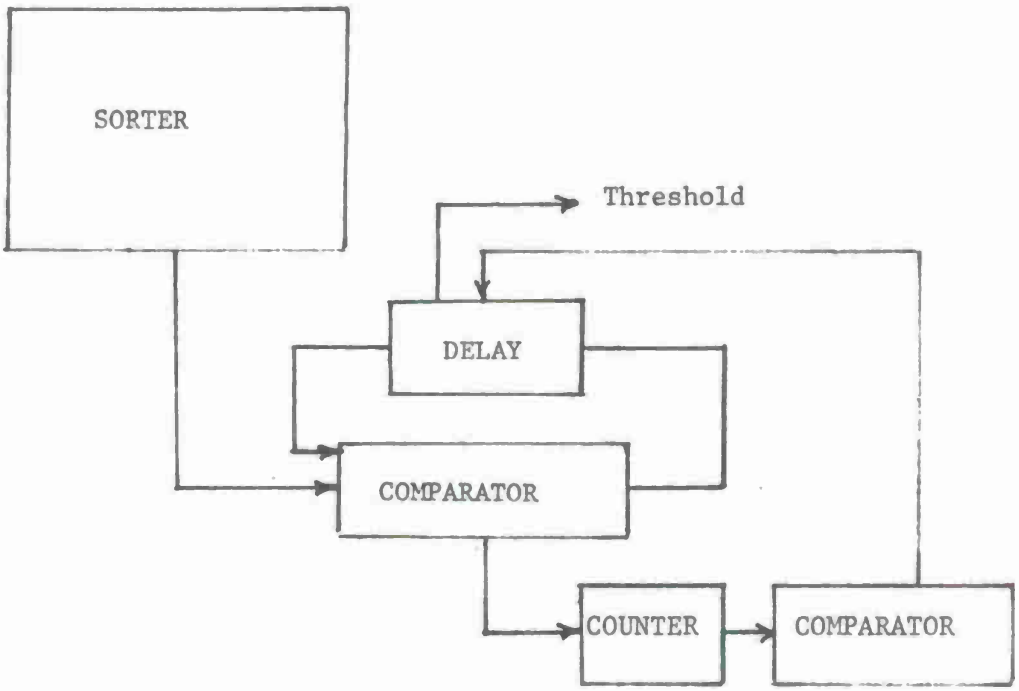


Figure 4.3-15. Threshold Determination

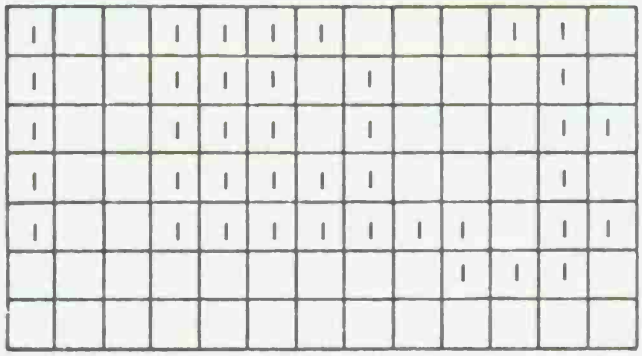


Figure 4.3-16a. Binary Image

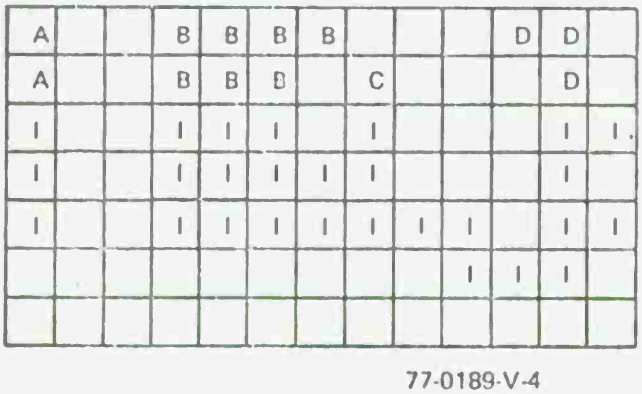


Figure 4.3-16b. Computations for second Row

Two image lines are retained in memory so that each pixel can examine its neighbors to the left and right and above and below. No diagonal connections are permitted under this convention, and an adjacent (horizontal or vertical) pixel must be occupied in order to make a connection. No skips or gaps are allowed, and the computations start one pixel in from the edge. In Figure 4.3-16b, there are four distinct regions, A, B, C, and D. The only possible connection between regions B and C is through a diagonal, which is not allowed. Computations for the fourth row are seen in Figure 4.3-16c.

A			B	B	B	B				D	D	
A			B	B	B		C				D	
A			B	B	B		C				D	D
A			B	B	B	C	C				D	
I			I	I	I	I	I	I	I		I	I
									I	I	I	

77-0189-V-5

Figure 4.3-16c. Computations for Fourth Row

Here, there is a connection between regions B and C and an equivalence statement, B = C, is carried along. At the end of the sixth row, there is another connection between C and D (C = D) and all the regions are completed as seen in Figure 4.3-16d.

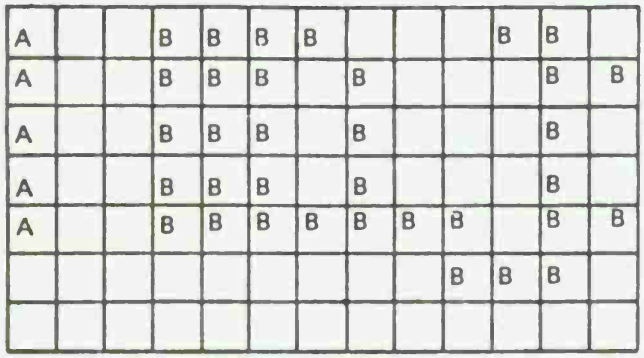


Figure 4.3-16d. Completed Image

The areas of A, B, C and D are computed by cumulating the number of pixels assigned to each. The perimeter is calculated by cumulating the number of pixels assigned to each region which are neighbors of zeros, i.e., the neighbors did not exceed the gray level threshold, g_i.

a. Coloring Operator

We assume that the Connected Components Algorithm processes a binary image, i.e., each pixel contains either a one (1) or a zero (0). The binary data stream will enter the Coloring Operator and emerge transformed into different colors or signal levels for different shapes. Read out of the binary picture will progress one horizontal line at a time starting with the top line and progressing downward. Each horizontal line will be read out from left to right. Since the image data is read out serially, the Coloring Operator is a local operator. The coloring of each non-zero element in the image plane will be done as shown in Figure 4.3-17.

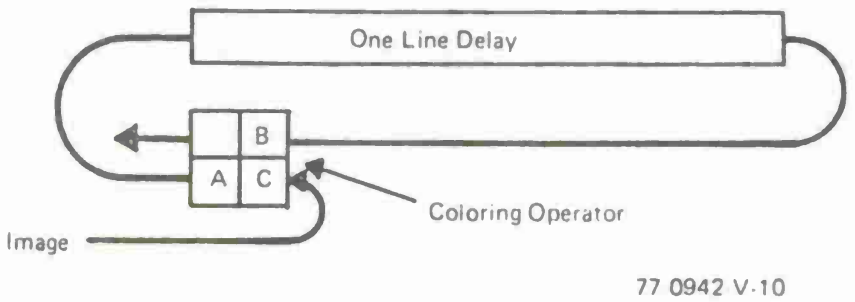


Figure 4.3-17. Data Flow Through Coloring Operator

The Coloring Operator is a transformation from a binary picture to a colored one by a mapping T

$$T\ (A,\ B,\ M):\ C \qquad C^1$$

where C^1 is the color of the transformed pixel C, the variables A and B represent nearest neighbors of C, and M represents an available color. The relative locations of pixels A, B, and C in the image plane are shown in Figure 4.3-18.

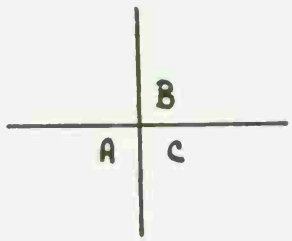


Figure 4.3-18. Relative Location of A, B, C

We define the coloring window as always containing these three elements. Elements A and B are nearest neighbors of C and have already been processed by the Operator. Element B is located one horizontal line above elements C and A. Element C is being painted by the Coloring Operator according to the following rule:

$$\text{For } C \neq 0 \qquad C^1 = \begin{cases} A \text{ if } A \neq 0,\ B \neq 0 \\ B \text{ if } A = 0,\ B \neq 0 \\ M \text{ if } A = 0,\ B = 0 \end{cases}$$

When adjacent elements have different colors, the element being painted assumes the color of the nearest neighbor in the same line (rows dominate). Whenever elements A and B are zero and element C is not zero, element C^1 is given a new color. Multi-colored object regions occur when pixels A and B are both

non-zero, not equal to each other, and C is non-zero, i.e.,

$$\text{For } C \neq 0, \quad A \neq B \neq 0$$

Painting of the binary picture with different colors will be accomplished by using a different number of charge quanta to represent various colors. The number of colors required is a function of target size, shape, and frequency of occurrence within the image plane. For purposes of the immediate discussion, we shall assume K colors are required, therefore each non zero pixel can require up to K memory locations to represent its color by a thermometer code (see Quantizer Section 4.3.1.2).

The configuration of the Coloring Operator is shown in Figure 4.3-19; note that there are K outputs C_K^1 for a single input C. Such a construction is required to reduce cummulative errors if analog implementation of Module C is attempted. Moreover, the configuration shown in Figure 4.3-19 is readily amenable to LSI and CCD technology. Each bit of C^1 output is governed by the expression,

$$\text{For } C_K^1 \neq 0$$

$$C_K^1 = \begin{cases} A_K^1 \text{ if } A_K \neq 0,\ B_K + 0 \\ B_K^1 \text{ if } A_K = 0 \\ M_K^1 \text{ if } A_K = B_K = 0 \end{cases}$$

The required input for operating the Coloring Operator are C, A^1, B^1 and M^1. Input A^1 is readily obtained by tapping the first vector element stored in the SI/SO CCD delay line. Similarly, vector element B^1 is also obtained by tapping the output of the SI/SO CCD delay line. The construction of this delay line is readily feasible and has been addressed in the Data Flow Section.

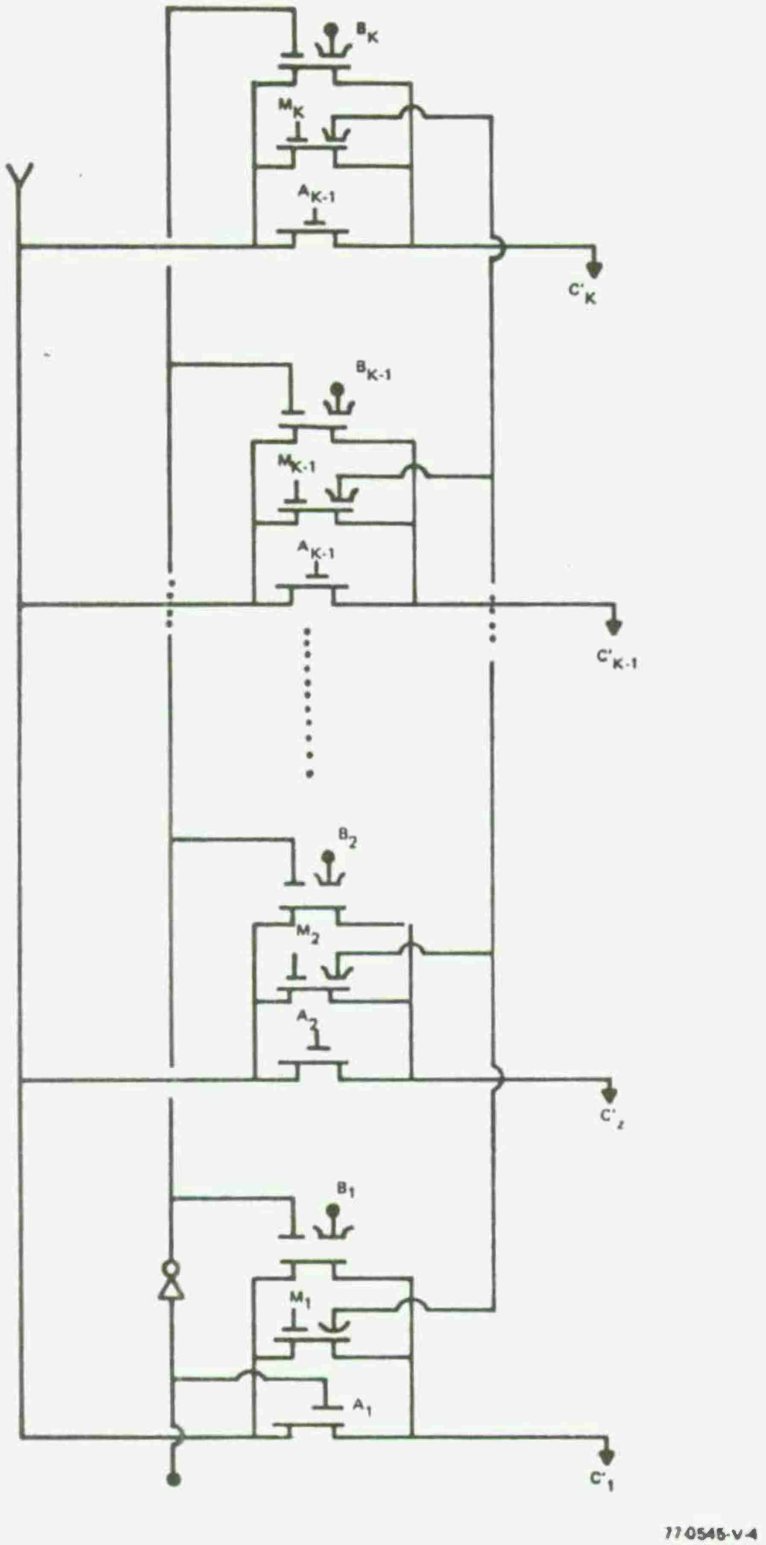


Figure 4.3-19. Details of the Coloring Operator

b. System Block Diagram

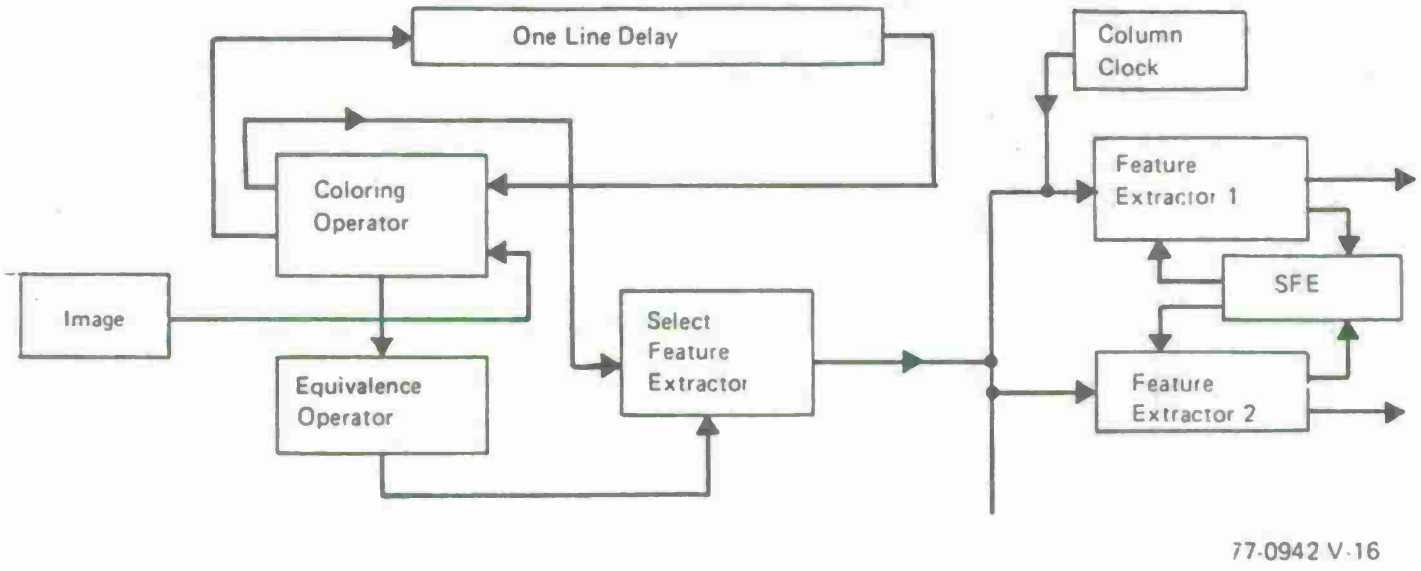


Figure 4.3-20. System Block Diagram

The System Block Diagram for the Connected Components Algorithm is shown in Figure 4.3-20. Before continuing with the specific hardware implementation it is useful to consider a general description of the implementation as a review and an indication of where we are going next.

The purpose of the Connected Components Algorithm is to segment an image frame into object regions; these object regions are potential shapes of interest and features are extracted from them for classification purposes. We assume that Time Delay Integration is part of focal plane signal processing which implies that the image comes to the cuer in the form of one line at a time, i.e., the pixels in one line arrive in parallel. The Connected Components Operator then moves along this line of pixels, with the previous line in memory, determining which pixels are part of a particular object region or if a new object region is starting. If we are to extract features from each object region, there must be a means for distinguishing between different object regions. One approach to the problem is to color each object region with a different color and then have a feature extractor operator assigned to each color. Where an object has several colors, the feature extractors corresponding to those colors accumulate their features, dump them in a scratch feature extractor to combine them, and reassign the result to one of the two feature extractors. Let us now proceed into the specific areas.

c. Feature Extractor

There is a Feature Extractor corresponding to each color as shown in Figure 4.3-20. The signals shown in Figure 4.3-20 enable the particular Feature Extractors for computations when an object has several colors. They also direct which Feature Extractor will receive the contents of the scratch Feature Extractor, i.e., which color will dominate. We have organized Figure 4.3-20 so that the number of inputs to the Feature Extractors is minimized. Each Feature Extractor is visualized as a many-channeled, large holding well which follows along the lines of the sorter. (See Section 4.3.2)

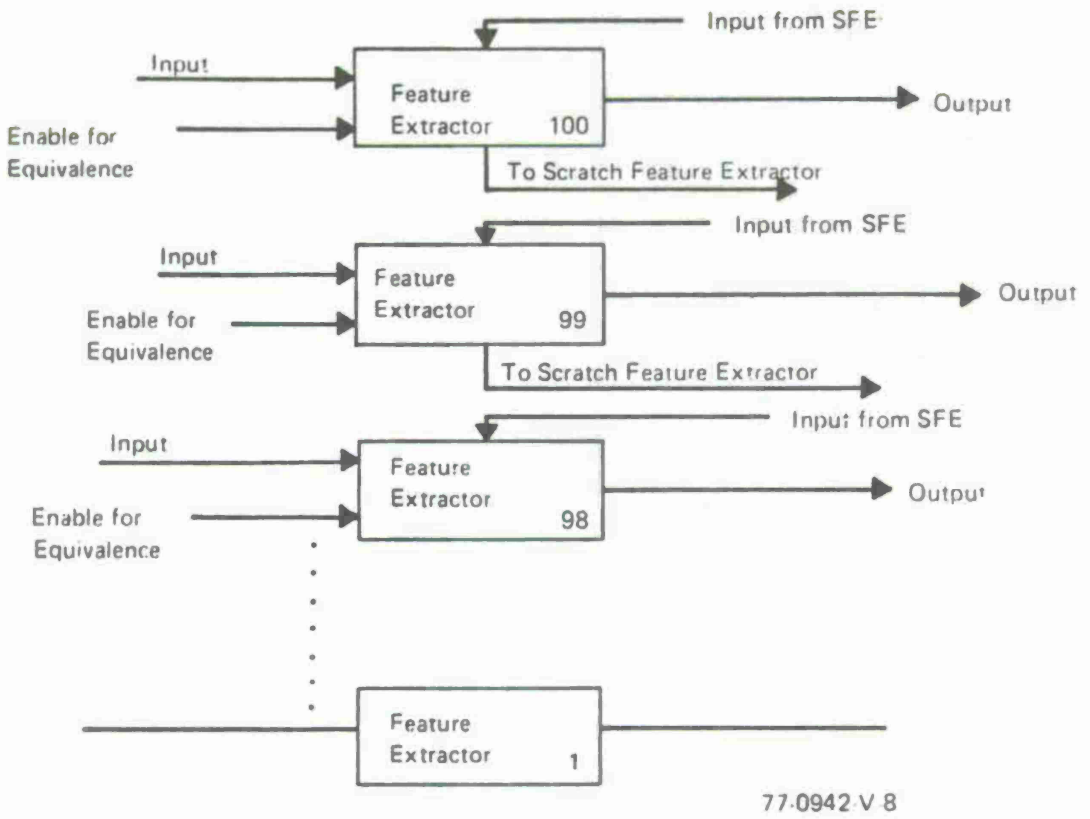


Figure 4.3-20. Organization of Feature Extractors

Each channel would correspond to a particular feature and since the features are cumulative, they would simply add in the scratch feature extractor. A discussion and implementation of the features by the University of Maryland is found in Section 3.8. We have described the front and back ends of the Connected Components Algorithm, now we lay out the middle portion which is designed to handle multi-colored objects.

a. Equivalence Statements

We present in Figure 4.3-21 an arbitrary object region which provides sufficient characteristics to explain the remainder of the Connected Components Algorithm.

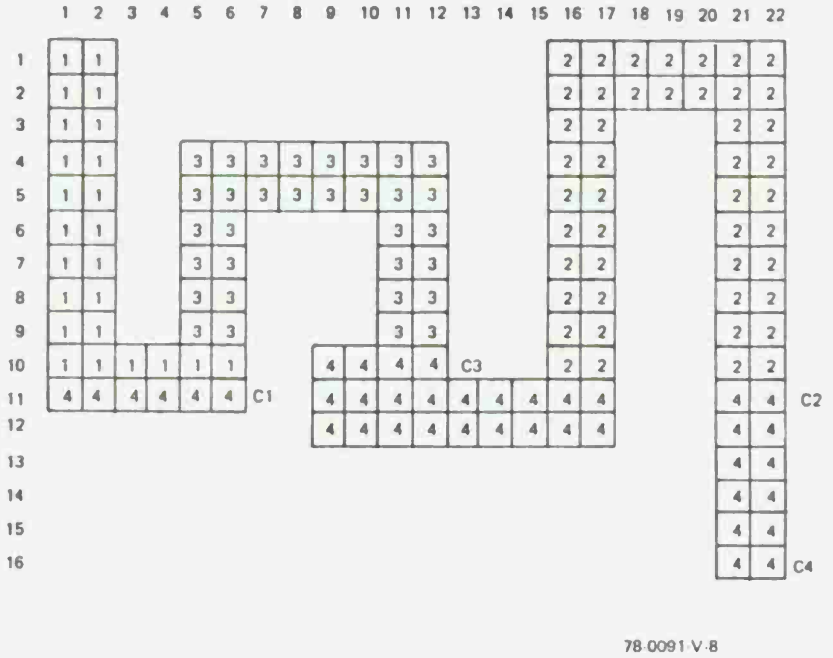


Figure 4.3-21. Arbitary Object Region

We are coloring the object from left to right and from the top to the bottom. At row 1, two seemingly different objects are identified and are colored 1 and 2, respectively. At row 4, a third object is detected which doesn't have any vertical or horizontal connections to objects 1 and 2 and so is colored with another color, no. 3. The coloring of these apparently distinct objects continues through row 9. Feature Extractors 1, 2, and 3 are accumulating features for

each of the three objects and their respective close out clocks are running. At pixel (10.5) a vertical connection between different colors (1 and 3) is noted. Since rows dominate, color 1 is placed in pixel (10,5). We are now alerted that Feature Extractors 1 and 3 will be combined in the Scratch Feature Extractor at some time in the future and we want to remember this equivalence. Since RAM structures are, so far, alien to the CCD world, we shall use a Switching Matrix as shown in Figure 4.3-22. A connection is formed between row 1 and column 3. Then every time color 3 is encountered

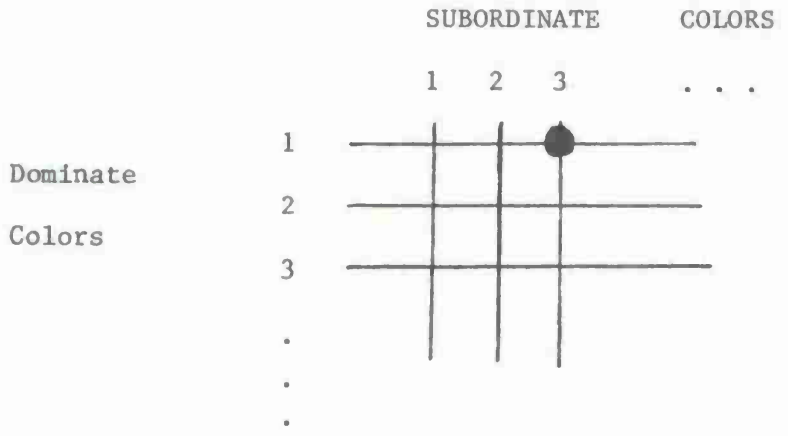


Figure 4.3-22. Switching Matrix

in the object region, the 3 line is pulsed and the signal exits the Switching Matrix on row color 1. At pixel (10,6) for example, we paint color 1 because there is a horizontal connection with pixel (10,5) and column color 3 leaves the Switching Matrix on the row color 1 line. At pixel (10,9), an apparently new object is encountered and color no. 4 is introduced. At pixel (10,11), a vertical connection between color 4 and 3 is detected and color 4 dominates. Color 3 is already latched to color 1, so the connection establishing color 4 as dominating color 1 is made in the switching matrix as shown in Figure 4.3-23.

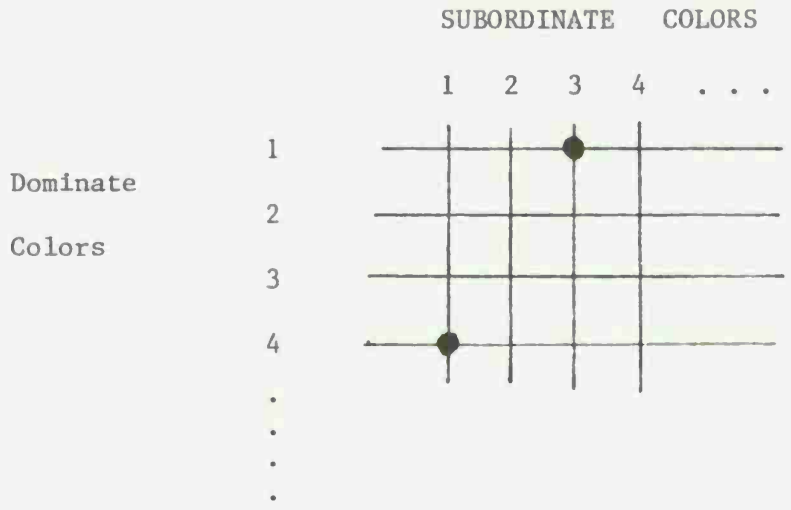


Figure 4.3- 23. Switching Matrix

Color 1 is a dominate color with respect to color 3 and a Subordinate Color with respect to color 4. At pixel (10,12), color 4 is painted in because of the horizontal connection with pixel (10,11). At pixel (10,13), color 3 is closed out as shown by the symbol C3 in Figure 4.3-21. This means that an entire horizontal image has been traversed since the last color 3 was entered. There is a clock associated with each Feature Extractor which keeps track of the time since the last entry to that Feature Extractor. The clock turns over at the rate that the pixels are processed as the Coloring Operator moves along a line of image. When the clock equals the number of pixels in a line of image, the color is closed out. To close out a color, e.g. color no. 3, Clock no. 3 sends a pulse to the Switching Matrix which identifies color no. 1 as the other part of the equivalence. Since color no. 3 is closed first, it will be merged into color no. 1. This is accomplished by entering the contents of Feature Extractors 1 and 3 into the Scratch Feature Extractor to combine them, and placing the results in Feature Extractor 1. Then the connection between colors 1 and 3 in the Switching Matrix is opened and color 3 is ready to be used again. The remainder of row 10 is straight forward.

At row 11, reference to the Switching Matrix places color 4 in pixel (11,1). Also at pixel (11,7), color 1 is closed out, the result placed in Feature Extractor no. 4, and the connection between colors 1 and 4 in the Switching Matrix is opened. Color 4 continues to be painted through row 11, including pixel (11,16) where row colors dominate column colors. And a connection between colors 4 and 2 is established in the Switching Matrix. Color 4 is painted at pixel (11,21) because of this connection. At pixel (11,23) color 2 is closed out and the results placed in the Feature Extractor of color 4. The remainder of the object region is straight forward. The Switching Matrix, connections (●) and disconnections ✱, and the equivalence statements are shown in Figure 4.3-24.

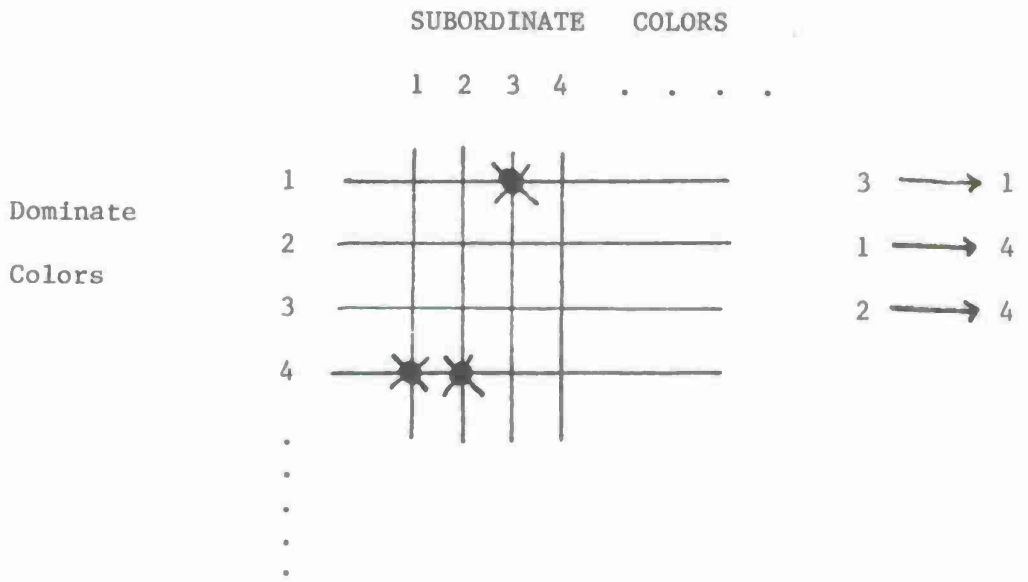


Figure 4.3-24.

e. Switching Matrix

The Switching Matrix is shown in Figure 4.3-25.

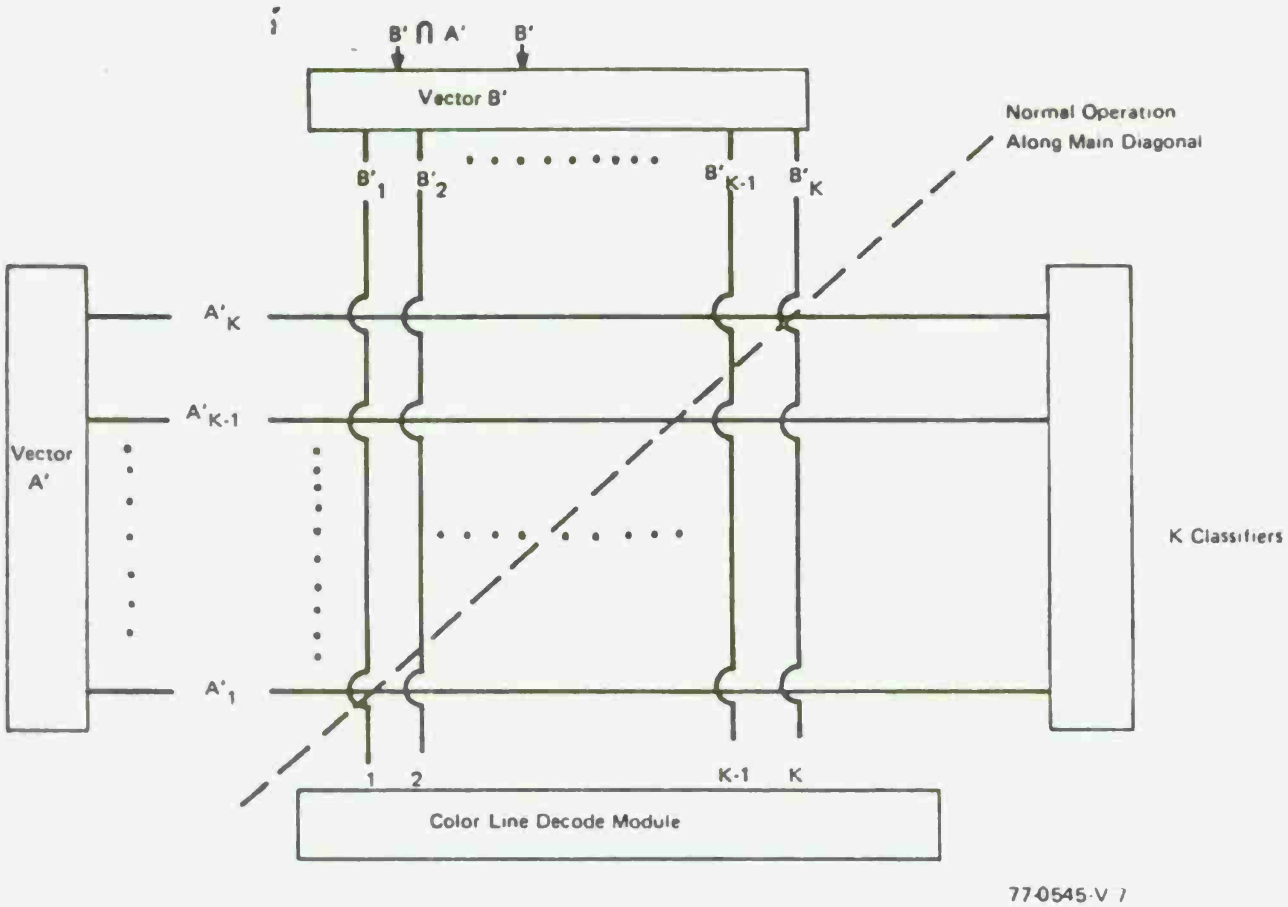


Figure 4.3-25. Switching Matrix

Each color data element, analog signal occupying one CCD site, is quantized into a K element vector. Only the highest subscript vector element is retained for each color. The highest element retained enables one vertical line of the K lines connected to the Color Line Decode Module. Each vertical line from the CLDM is connected via a switch to only one horizontal line going to the K Feature Extractors. Normally the K^{th} CLDM line is connected to the K^{th} horizontal classifier line.

Quantizing the analog color signal requires a Quantizer described previously in regard to the Median Filter. Enabling the particular vertical line to the CLDM is not difficult since the largest non zero element is followed by a zero element unless it is the K^{th} element, in which the K-1 and K^{th} elements are both ones. Figure 4.3-26a shows an enabling circuit for the vertical CLDM lines.

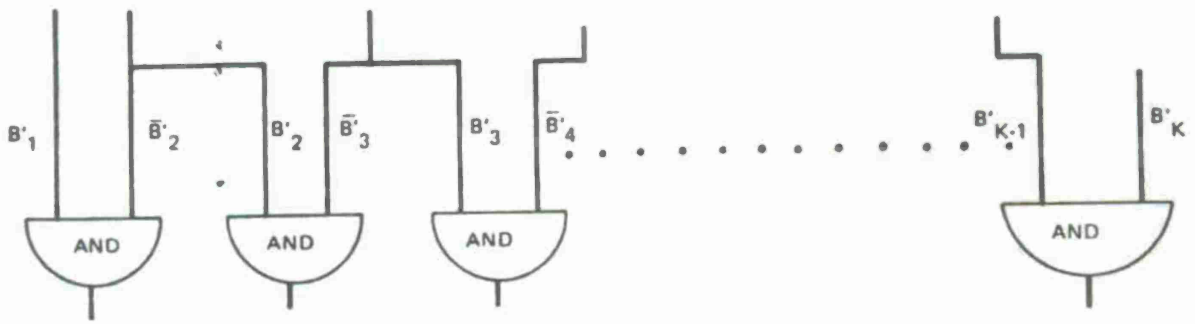


Figure 4.3-26a. Enable Circuit

The color element corresponding to the vertical CLDM line is fed directly to the AND gate. The complement of the next higher element is also fed to the AND gate; this gives the non-zero, zero combination. For the K^{th} line, the K-1 element is fed uncomplemented to the K^{th} AND gate. This is the normal mode of operation for the Switching Matrix; next we consider the case when portions of the same target have different colors.

We will show how the outputs from vector A^1 and B^1 modules can modify the diagonal interconnection between the CLDM and the K Feature Extractors. A switch is located at each intersection between the horizontal and vertical lines in Figure 4.3-25. The function of this switch is to repaint each target into a single color and this switch will be reset when the color has been closed out. For example, if vector A^1 has color i, $1 \leq i \leq K$, and vector B^1 has color j, $i \leq j \leq K$, $i \neq j$, only the i, j latch switch is enabled. Once this happens, all the data elements to be painted with the j^{th} color will be painted with the i^{th} color. The i and j elements will be channeled to a single Feature Extractor i, also. This color merge operation is achieved by connecting the i and j outputs from the CLDM unit via switches to the i^{th} Classifier.

To a large extent, then, merging different colors within the same target can be achieved by selecting appropriate switches and connecting them at the intersection between the vertical and horizontal lines of Figure 4.3-25. A diagram of such a switch is shown in Figure 4.3-26b. Two types of switches are required for the Target Sorter; switches on the main diagonal will be different from those placed off the main diagonal. From the switch configuration shown, only one switch will be on in any given column. Therefore, at most only one Feature Extractor line will be connected to each vertical line.

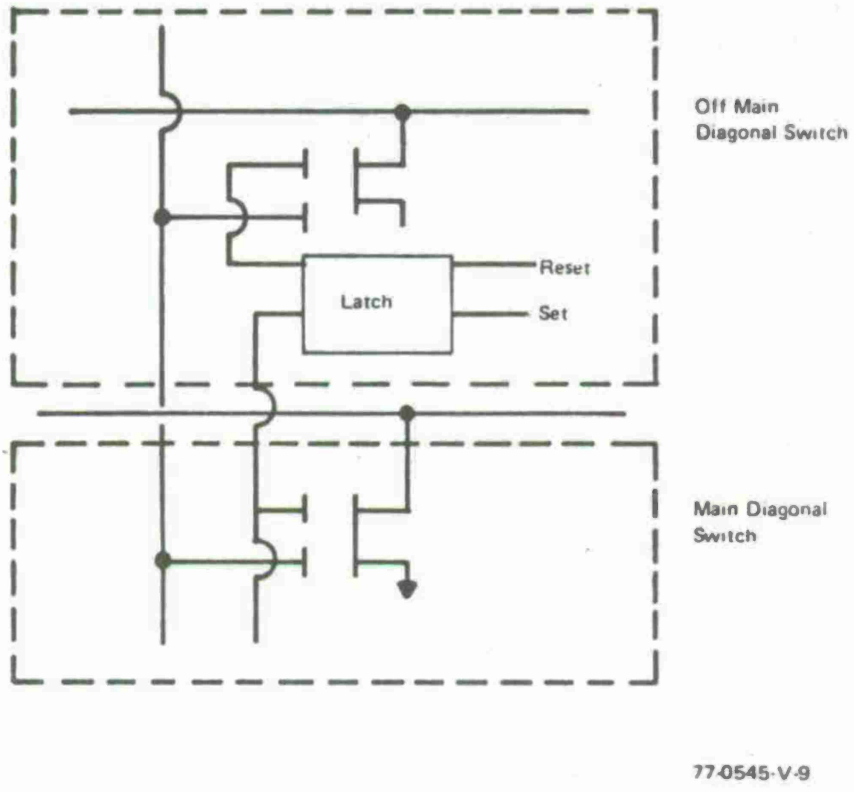


Figure 4.3-26b. Switches within the Target Sorter

The latch shown in Figure 4.3-26b is reset such that initially only the main diagonal switches can be enabled. A reset input to each i, j latch switch will be controlled by the outputs from the vector A^1 and B^1 modules shown in Figure 4.3-25. Each module will only contain one element with the highest subscript like the vector in the CLDM module. Existence of non-zero elements in the vectors A^1 and B^1 modules will only occur when the Coloring Operator detects two adjacent colors during the target painting. That is, when $A^1_{\cdot} \neq B^1_1 \neq 0$ a condition which represents multi-color in one target. A flag for such a condition is provided by the CO and will enable the A^1 and B^1 modules for one clock period. The set inputs to the latch switches are gated such that only one latch is enabled for any combination of vectors A and B.

Once the j^{th} color has been closed out, it will reset the j^{th} line such that all latching switches connected to the j^{th} line (excluding the j^{th} CLDM switch) will be open. Moreover, the j^{th} color will be returned to the Color Storage and Memory Module located in the Color Operator Unit.

4.3.6 Super Slice

Suppose an image has been processed through the Non Maximum Suppression and Connected Components Algorithms independently as seen in Figure 4.3-27.

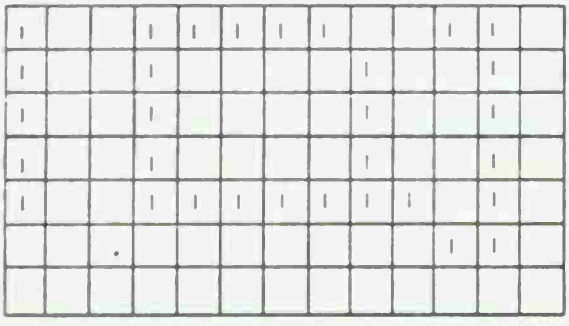


Figure 4.3-27a. Non Maximum Suppression Output

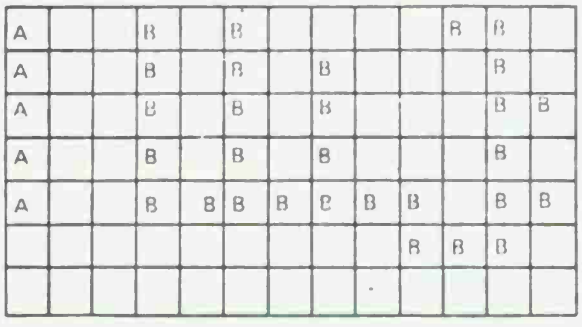


Figure 4.3-27b. Connected Component Output

Region A has a perfect score: every pixel of maximum gradient is matched as a perimeter point of A. Region B matched 23 points out of 26 possible gradient points but it also produced 5 perimeter points which were not matched by gradient pixels. This match is done, in parallel, for each of the threshold levels.

4.3.7 Feature Extraction

A tentative list of features which Maryland will use for classification includes perimeter match, area, perimeter extent, average gray level, and maximum height and width. The purpose of this section is to didcuss the implementation of a feature extractor.

At this point in the data flow, we may assume that the object has been segmented and the existence of more than one color within the object has been resolved. Also the object in its entirety has been sent to one of the K Feature Extractors discussed previously. Maryland has produced a tentative set of features thus far so the hardware effort will be limited, at present, to those features. We shall continue with the assumption that the image is entering the Feature Extractors one horizontal line at a line, the processing is moving down the image, and from right to left.

a. Area

Assume an object shape such as that shown in Figure 4.3-28.

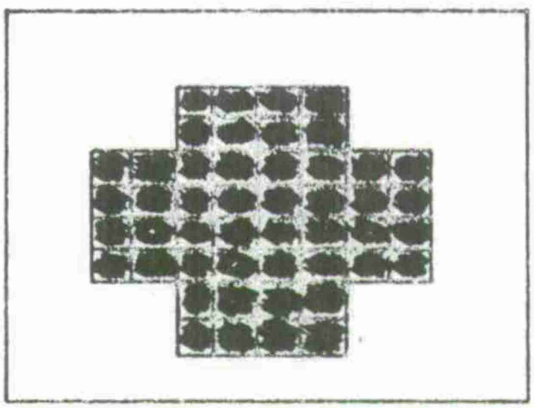


Figure 4.3-28. Arbitrary Target

The area of the object can be obtained simply by summing the number of pixels within the target; in this case the area is 48 pixels. The Area Feature Extractor then is seen in Figure 4.3-29.

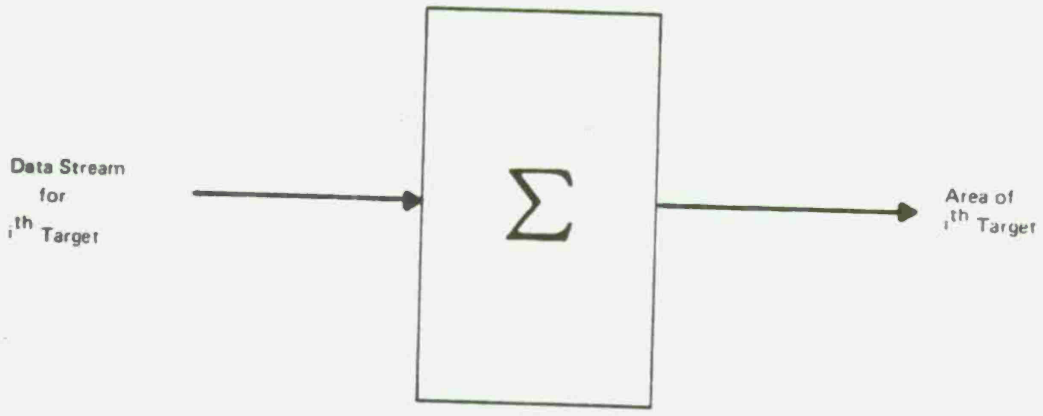


Figure 4.3-29. Area Feature Extractor

b. Perimeter

Both the Perimeter Match and Extent may be considered together. Each pixel has eight neighbors; a perimeter point is defined as having at least one zero neighbor, similarly an interior point is defined as having eight non-zero neighbors. Recall that these definitions apply to the binary image which was developed prior to the Connected Components Algorithm. However, in the Classifier Module no exterior neighbors would be allowed to enter, so the test of a perimeter pixel may be conducted as the complement of an interior target pixel. That is, only target pixels enter the Classifier; these target pixels which are not interior target points must be perimeter target pixels.

The Perimeter Feature Extractor would consist of three lines of memory, each 525 pixels long corresponding to the image frame width. The center pixel would be the location of the pixel being examined for neighbors.

An interesting problem is horizotally aligning these three lines of memory within the Classifier in the same way they are aligned in the image. This can be accomplished by using three lines of memory forming a serpentine and injecting a bit into the delay lines for the appropriate color location. Non destructive readouts form the 3 x 3 moving neighborhood as the pixels shift through as shown in Figure 4.3-30.

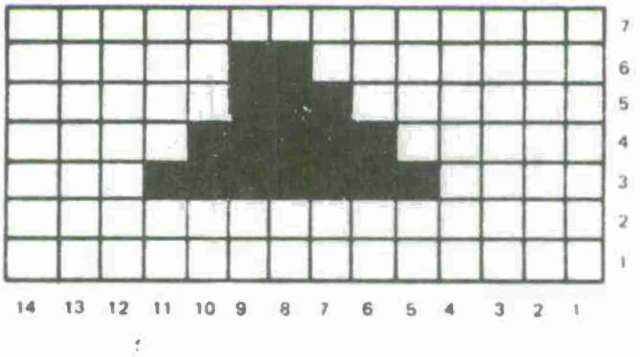


Figure 4.3-30a. Image

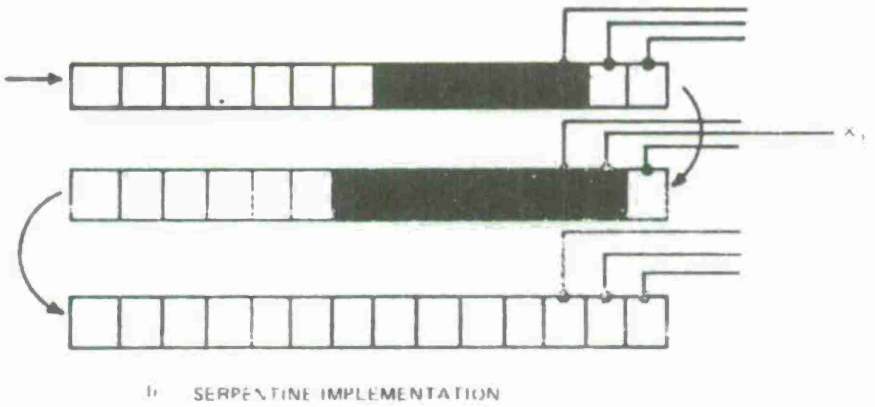


Figure 4.3-30b. Serpentine Implementation

The second to last pixel in the second row, X_T, is always the pixel position being tested as a perimeter point. This implementation does not have to make special allowances for pixels located on the edges of a target. This is the same technique described in the second and third quarterly reports for forming the moving windows for Gradient Operator, Median Filter, and Non Maximum Suppression. We can AND the neighbors of X_T, complement the output and AND it with X_T. If the output of the first AND is zero, then one of X_T's neighbors is zero and X_T is a perimeter point. See Figure 4.3-31 for the logic. This is fed to a summer and the total number of perimeter points for a target is produced. For the Perimeter Match, the important thing is to clock the

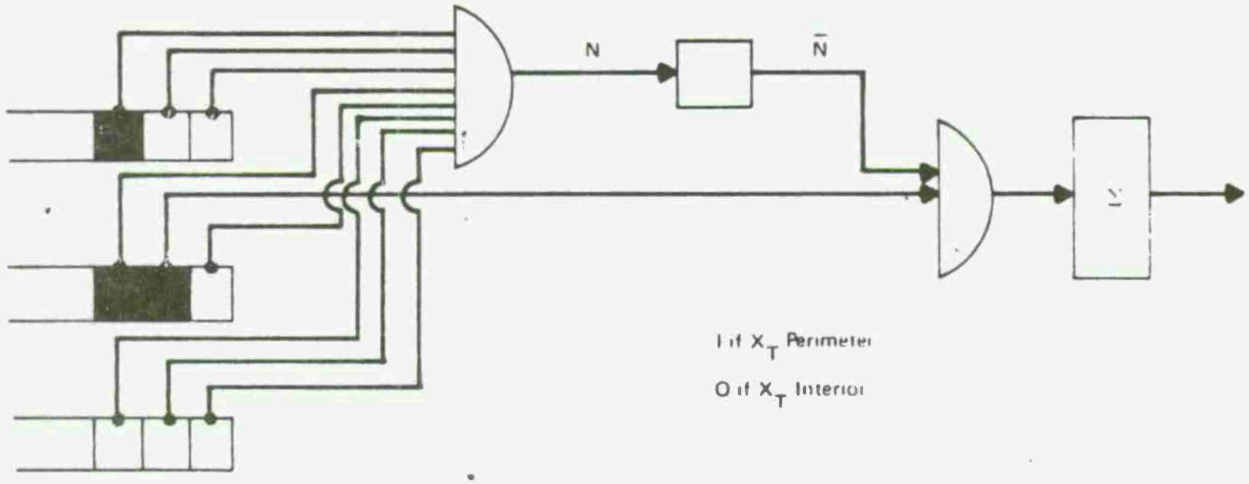


Figure 4.3-31. Perimeter Point Logic

image of edges at the same rate that the perimeter extent of the same target is determined. Then X_T is compared with its geometric counterpart in the edge domain, i.e., at the same x, y position in the image. Both are fed

to an AND gate and if non-zero, will increment another summer which keeps track of the Perimeter Match Score.

c. Maximum Target Height and Width

This has to do with keeping track of the maximum target excursions in x and y. For the maximum x extent suppose we use the same three delay line memory which was used for Perimeter Match. We further assume a more complicated object as shown in Figure 4.3-32. The left most site of the second row and the complement of the right most site of the last row are ANDed. If a one comes from the AND gate the summer is incremented by one. Essentially, we count the number of pixels in the first row of the image and then determine the amount of extension in the following rows and add it to the first sum. Note that number indicated by the summer corresponds to a count of the number image columns in the first rows.

The y extent of the image can be determined by counting the number of pixels passing through the three line delay and dividing the number of pixels per line.

d. Average Gray Level

This involves clocking the Median Filtered image at the same time the color elements are clocked through the Feature Extractor. The gray level sum is obtained and divided by the area obtained earlier by the Feature Extractor. The quotient yields the average gray level.

4.3.8 Data Flow

The Median Filter, Gradient and Non-Maximum Suppression Operators are calculated for small windows which move over the entire frame. These windows are formed by parallel shifting one line of image from the TDL array into a parallel in, Serial out Shift register (Figure 4.3-33a). This register and others then forms a serpentine delay through which the pixels are shifted.

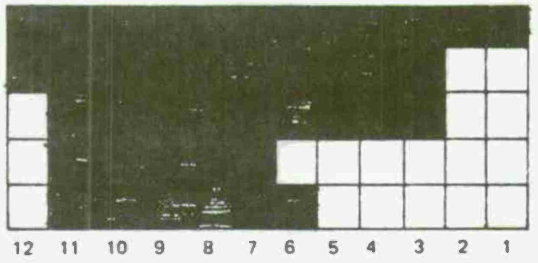


Figure 4.3-32a. Object

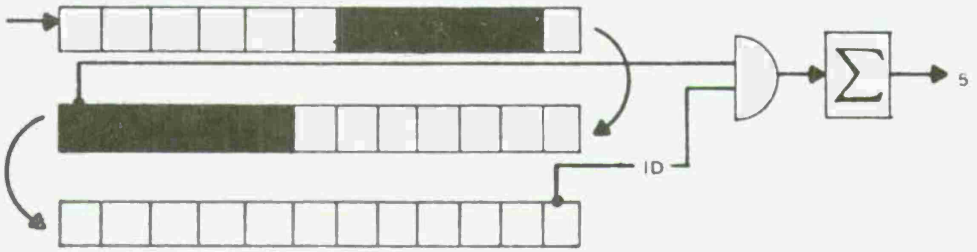


Figure 4.3-32b. Tapping First Image Row

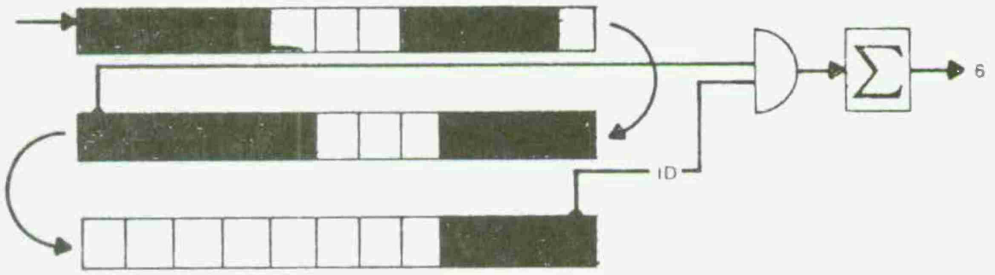


Figure 4.3-32c. Tapping Second Image Row

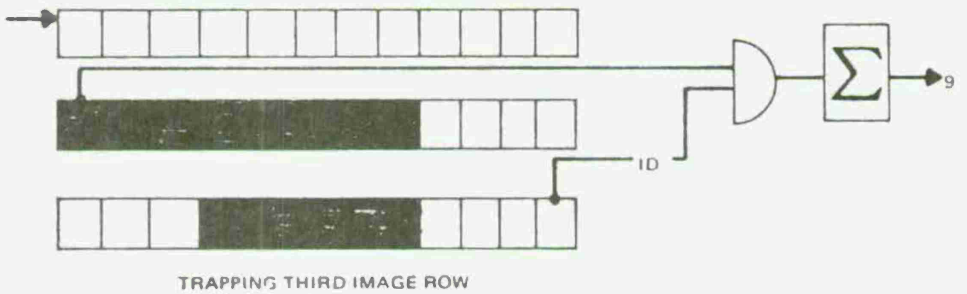


Figure 4.3-32d. Tapping Third Image Row

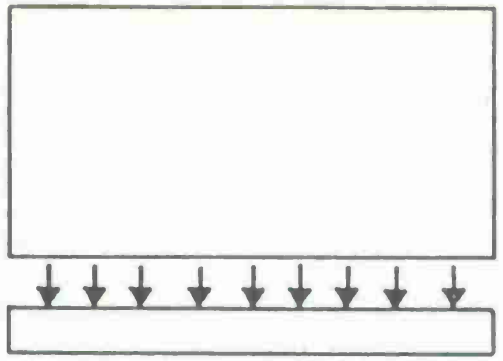


Figure 4.3-33a. Focal Plane Arrangement

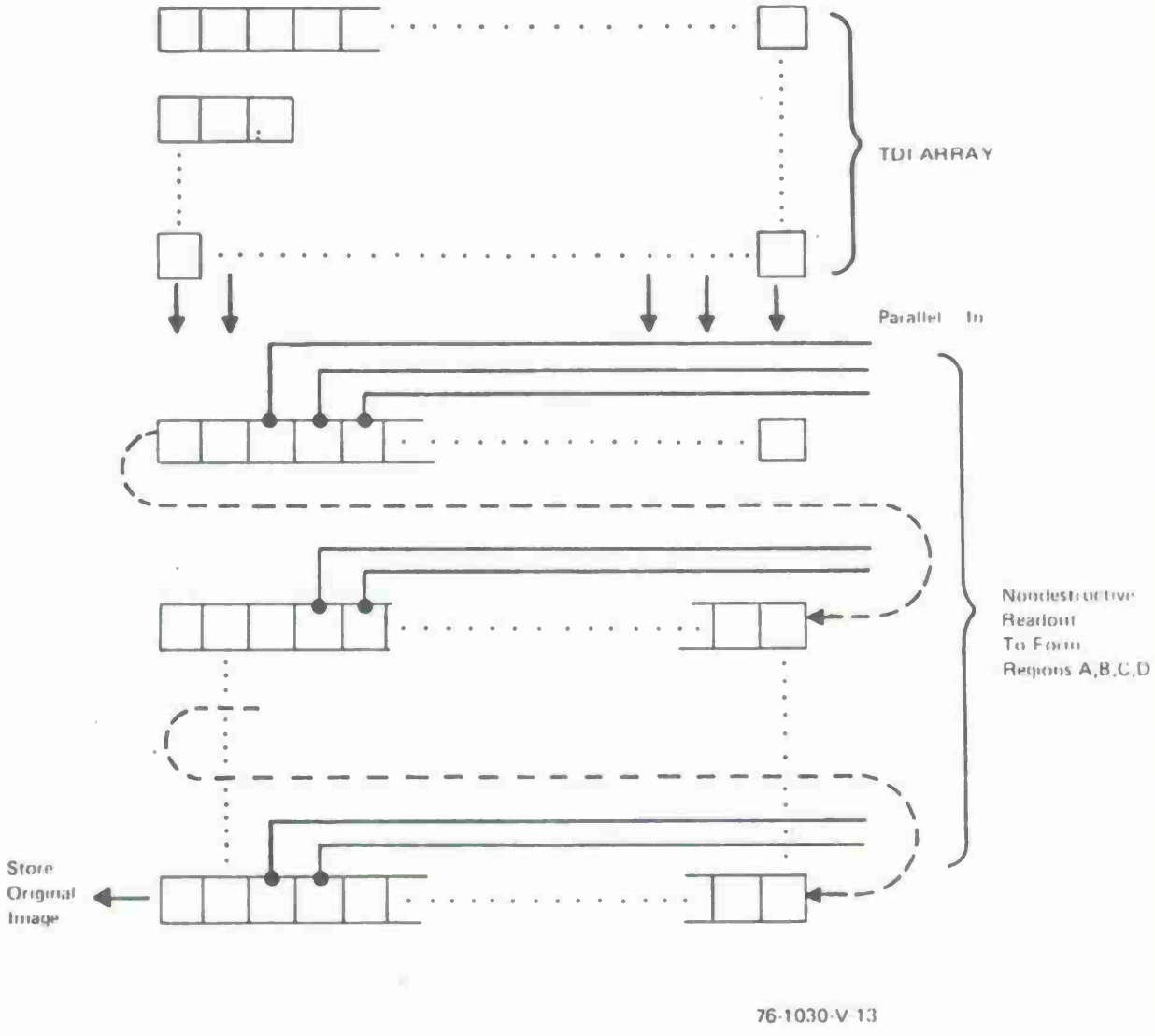


Figure 4.3-33b. Details of Serpentine Delay Line

Non-destructive readouts form the regions comprising the appropriate window (Figure 4.3-33b).

It appears that the computation speed of the Median Filter and Gradient Operator is conservatively estimated at 50-100 kHZ, hence a parallel organization of the focal plane (see Figure 4.3-34) is necessary for a 1 megapixel/sec. data rate. Suppose we divide the PI/SO register immediately below the focal plane into 20 vertical sections, each approximately 34 pixels wide, and each with its own serpentine CCD delay line as seen in Figure 4.3-35. If the image is 640 pixels wide, we divide the register into 20 sections of approximately 34 pixels each to avoid problems associated in calculating medians and gradients along the edge of an image. Each vertical section is eight stages long to accommodate the Gradient Operator, which requires eight lines of storage, and 34 pixels wide for a total number of 272 shifts at 50 kHZ. This appears to avoid numerical integrity problems. Performing the three algorithms consecutively requires three separate moving windows and clocks to control the non destructive readouts for each. The Median Filter Algorithm requires 5 stages of delay, the Gradient Operator requires 8 lines (stages), and the Non Maximum Suppression Algorithm requires 7 lines. The series arrangement is shown in Figure 4.3-35. The outputs from Non Maximum Suppression feed the SuperSlice Algorithm. Referring to the System Flow Chart (Figure 4.3-1), the other path requires that the original image be thresholded at g_1, g_2 ... g_n values. The computation of g_1, g_2,...g_n will require some knowledge of all the gray level values within a frame, hence frame storage can be expected, or thresholds determined from the two prior frames will be used for the present frame. In any event, having

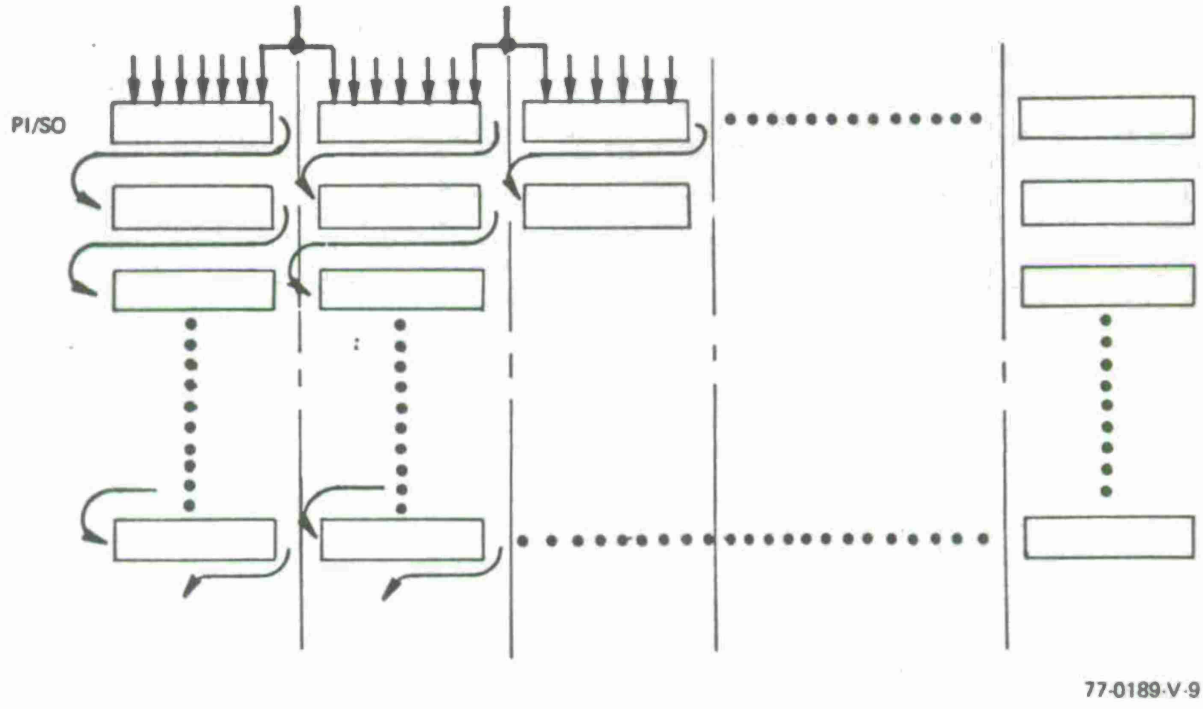


Figure 4.3-34. PI/SO and Serpentine in Parallel

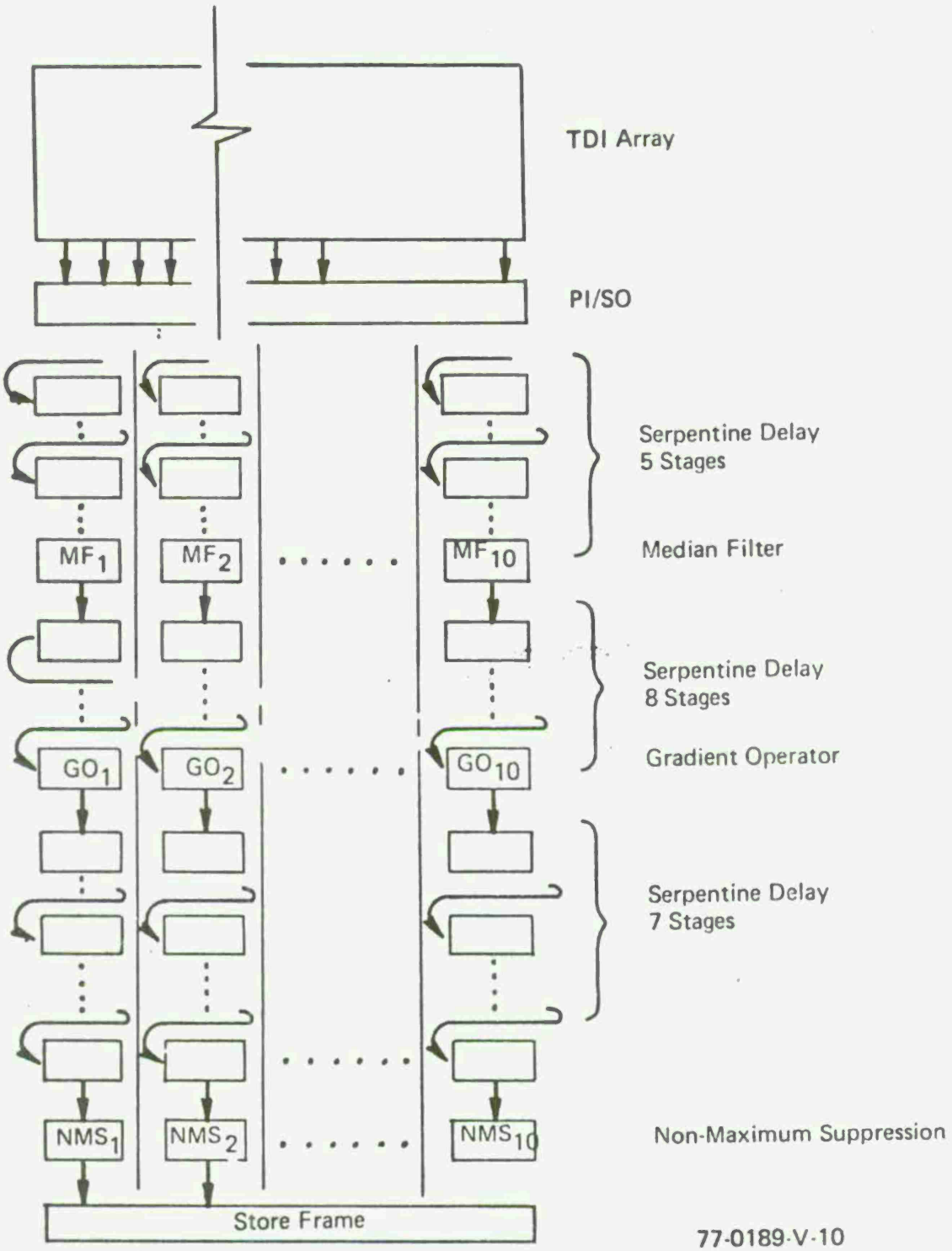


Figure 4.3-35. Focal Plane Data Flow

obtained g_1, g_2, ... g_n, the entire frame can be thresholded, by examining the rows in parallel, and Connected Component analysis started, as in Figure 4.3-36. The frame is clocked out of storage, one pixel at a time, and each pixel is thresholded in parallel forming frames Fg_1, Fg_2, ... Fg_n. Similarly, the Connected Component Algorithm is performed in parallel with the required one line delay for each. To avoid additional storage, the SuperSlice Algorithm will be done as the output from Connected Components becomes available and the scores cumulated.

4.3.9 Storage Requirements

From Figs. 4.3-34 and 4.3-35, it appears that 20 stages of serpentine delay are necessary, each stage is approximately 34 pixels long and divided into 20 separate sections for a total of 400 delay lines. To achieve a 1 megapixel/sec. speed, we are conservatively talking about 20 Median Filter Processors, 20 Gradient Operator Processors, and 20 Non-Maximum Suppression Operators for a total of 60. Assuming five thresholds, as an example, an additional 12 lines of delay are required, plus five Connected Component Processors.

4.4 Hardware Fabrication

4.4.1 Algorithms

In previous work, we have described the hardware implementation of a number of algorithms. In this section we shall take that work a step further and consider the fabrication of these implementations. Specifically, we shall consider chip size, cryogenic problems, speeds, yields, and power consumption relevant to the Gradient Operator, the Median Filter, Serpentine Memory, Non-Maximum Suppression, and Connected Components Elements.

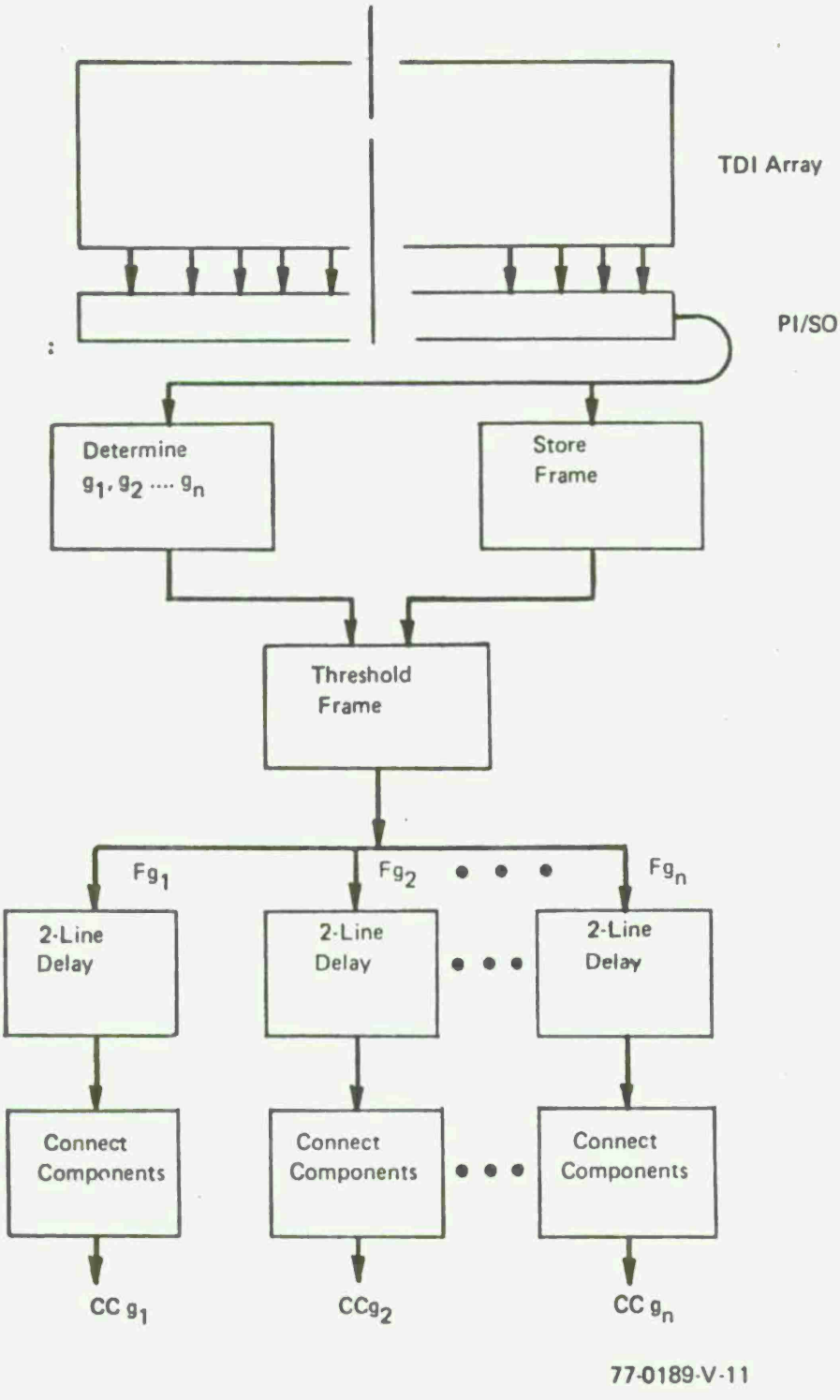


Figure 4.3-36. Connected Components Data Flow

a. Gradient Operator

A major assumption in the analysis is that the inputs for the Gradient Operator (i.e., A, B, C, and D) will be obtained from a separate IC chip which will be a part of the serpentine memory module. The structure of this module will be addressed in another section.

The size of the Gradient Operator chip will be deduced by assigning real estate to each operation performed by the Operator. A key operation is the absolute subtraction module (ASM) which obtains the absolute difference between two inputs and yields a charge representing that quantity. Each difference CCD structure will nominally require a channel 1.2 mils wide; four input channels are needed to provide four charge packets, two representing $|A_i - B_i|$ and two representing $|C_i - D_i|$. The length of each ASM will be 4 mils, a size sufficient to provide a readout structure necessary to drive the second stage of the Operator. The second stage selects which output $|A_i - B_i|$ or $|C_i - D_i|$ is the largest gradient of the i th pixel location. Combining the real estate requirement for the first and second stages, we calculate a chip size of 8 mils x 10 mils.

We assume a four phase gate construction; a smaller number of phases (which requires less chip area) could be used; however, speed-charge handling capacity and ease of fabrication favors four phase construction.

The structure advocated is exclusively based on MOS FET and CCD technology. Both MOS FET and CCD structures exhibit improved performance at cryogenic temperatures greater than 30°K. At very low cryogenic temperatures ($\geq$ 30°K), the performance of MOS CCD structures begins to show significant degradation. Relative to room temperature performance, experiments have shown that with cryogenic temperatures we should obtain higher operational speeds and lower noise figures. The improved performance is attributed to increases in mobility resulting from lower levels of phonon scattering of the signal carriers.

The fabrication yield depends on chip size and the number of steps. The process is very similar to that for making surface channel CCD and we estimate six photolithographic masks. Cognizant of these similarities, we expect a yield of better than 50%. The variables which will influence the final chip configuration will be speed, charge handling requirements and resolution (uniformity). Our present design is conservatively aimed at a speed of 100 kHZ. Higher speeds are possible, but increasing the operating speed from 100 kHZ to 1 MHZ will require special and more difficult structures.

Power consumption of the Gradient Operator will depend on the operating frequency and checking voltages. Conventionally, the power consumed by a CCD type structure is expressed as

$$P = CV^2 \, f \, N$$

where N is the number of gates, C is the capacitance of each gate, V is the clocking voltage and f is the operating frequency. Computing the power requirements we obtain less than 10 milliwatts. This level of power consumption is exclusive of the power requirements of the clocking circuitry required to operate the Gradient Operator.

b. Median Filter

In the second quarterly report, Maryland reported on the significance of the Median Filter Operator and Westinghouse described an embodiment. In this section, we shall consider aspects pertinent to fabrication.

The MFO chip as considered below will not include peripheral clocking circuits or a structure for summing the output from the serpentine CCD delay. We assume an MFO operating on 25 pixels located within a moving window; provisions for obtaining the 25 pixels will be built into the CCD serpentine delay structure in the form of non-destructive readouts. Each data element (pixel) is assumed to have a dynamic range equivalent to a 32 level grey scale.

The size of the chip is determined primarily by the number of pixels and grey levels. The proposed MFO is required to operate as a moving window device which requires a CCD memory capable of storing and shifting 25 data elements each of which is quantized within a 32 level grey scale. A bank of CCD memory registers with 25 x 32 storage locations can be achieved by a 64 mil by 64 mil module. Included in this estimate are areas for incorporating output and input structures to the CCD memory.

Another major block of the MFO is the sorting module in which the data elements are arranged according to size. This requires a bank of 32 CCD shift registers which are 25 elements long and each row is capable of being independently shifted left or right. An area 100 mils wide by 64 mils long is sufficient.

Finally, the area required for controlling the clocks operating the sorting module is estimated to be 100 mils by 2 mils.

Summing the different component areas comprising the MFO, we arrive at an area estimate of 100 mils by 128 mils.

All the elements used in modelling the MFO are based on field effect phenomena, hence we expect improved performance at cryogenic temperatures in accordance with experimental observations. As described in the Gradient Operator section, we expect improvement in bandwidth and noise reduction.

The size of the MFO (128 x 100 mils) represents a large scale integration device and significant complexity will be encountered during fabrication and test. We calculate that eight mask levels will be required. The yield, dependent on chip size and the number of masking steps, is estimated to be 5%.

Required power will be larger than that consumed by a conventional CCD device; the demand for more power comes from the active logic devices used for clock control (shift left or right) functions. Generally, power consumed depends on operating speed, component cost, and system layout. Assuming an operating speed of 100 kHZ, the MFO will require less than 100 miliwatts of power.

c. Serpentine Delay

A large number of delay elements are required for focal plane processing; the elements must have large memory capacity, good transfer efficiency, and non destructive readout structures.

The serpentine deployment shown in Figure 1-6 requires a large number of transfers which causes degradation in the modulation transfer function (MTF). This negative effect can be reduced by segmenting the image into several columns, each of which will be processed in parallel with the other columns. Such a segmentation not only reduces the number of CCD transfers per delay element but reduces the operating speed. We postulated an algorithm operating speed of 100 kHZ based on hardware implementations. This led to a division of the IR image into 10 columns, each 68 pixels wide. From Figure 4.3-36, it is seen that Median Filter requires five (5) lines of delay, Gradient Operator requires eight (8), and Non Maximum Suppression requires seven (7) for a total of 20, since the algorithms operate sequentially. The total number of shifts is 1360 per column. At a clock frequency of 100 kHZ, numerical degradation in the order of 20% will occur, which is probably too high. The MTF can be reduced in several ways.

The modulation transfer function is a function of the input signal frequency, the frequency of the shifts (clock frequency), the number of shifts, and the transfer efficiency. The more practical avenues of reduction are clock frequency and the number of shifts; we can double the number of operators to 20 each, and halve the clock frequency and number of shifts to 50 kHZ and 680, respectively. This may produce an improvement to 10% degradation, but this number would have to be confirmed experimentally. Of course this

approach increases the total chip area which is still small and the external clocking circuitry. Operating at cryogenic temperatures will probably increase the transfer efficiency somewhat. Further, the input frequency can be band-limited to decrease the MTF. Using the 20 column segmentation, each 34 pixels wide, a total of 680 shifts are required to perform the Median Filter, Gradient Operator, and Non Maximum Suppression Algorithms at a 1 megapixel/sec. rate.

Moreover, surface channel CCD's are suitable for this task within the defined operating parameters, and the advantage of these devices is realizing the non destructive taps. These taps are necessary in extracting the appropriate pixels for the moving windows discussed in Section 1.3 and the second quarterly report.

The size required for achieving a memory 680 elements long is 1000 square mils if four phase clocking is employed. Hence for 20 columns we will require a silicon area 1000 mils long by 20 mils wide.

Operation of the memory at cryogenic temperatures will present no problems since its construction is similar to the other focal plane signal processing components. Considering the size of this memory chip we expect a yield of about 5%. The clocking circuits required for the memory module operation are not included in the area calculations.

General Observations

In this report we have considered the physical parameters of the focal plane signal processing circuit elements. In our opinion, no signal processing operation defined and discussed contains any inherent characteristics which will prevent fabrication. However, the number and size of the required IC

modules is considerable. Integration of all the aforementioned elements in a single large integrated circuit is a high risk effort. Development of each single IC block first should provide sufficient test vehicles and data needed to evaluate each signal processing component. Such data should be obtained before any large scale integration of all the focal plane signal processing is undertaken. This approach will result in the most efficient method leading towards LSI focal plane signal processing.

d. Non-Maximum Suppression Operator

Estimating the physical size of the NMSO will be consistent with the design rules followed for estimating the fabrication of primitive operators reported in previous quarterly reports. It should be restated that the estimates provided are preliminary and adjustments are expected when chip layouts are made. Estimation will be made by first decomposing the operator into its components and calculating the physical dimensions of each port.

The NMSO is made up of several components. A serial input/parallel output CCD structure will be used in quantizing the analog signals injected into the NMSO. The size of this component will be 5 mils x 100 mils wide. The output from the quantizing module will go into a sorter which will be 13 mils by 100 mils in size. The output from the sorter will be injected into a CCD compare register which will compare two signals. One signal is the largest, x, of the neighboring gradient pixels; the other is gradient pixel of interest, y, (see Figure 1-2 in the Jan. 31, 1977 report). The difference output will control the NMSO output according to the following rule

$$\text{NMSO Output} = \begin{cases} 0 & y \leq x \\ y & y > x \end{cases}$$

Totaling the area required for each NMSO component, we obtain an area of 100 mils by 25 mils. This chip will have MOS structures. We estimate a 25 milliwatt power drain by the NMSO neglecting peripheral clocking and logic. A 10% yield for this chip is predicted.

e. Connected Components Algorithm

The Coloring Operator consists of several blocks; the coloring logic, the color delay line, the equivalence operator, and the color bank. Fabrication estimates have been discussed and obtained for these elements in the fourth quarterly report based on ten (10) colors. We now extrapolate those estimates to handle one hundred (100) colors.

Our previous estimate for the coloring logic was 10 mils by 50 mils. Laying these modules side by side and end to end, yields an area of 100 mils by 50 mils neglecting space for connections. The logic will be MOS structure to maintain compatibility with other structures. A color delay line will be 20 elements wide by one horizontal TV line long. Consistent with the area required for previously described serpentine memory chips, we predict a 400 mil x 300 mil size. The color bank will be very similar to the Median Filter already described. Accounting for the number of quantization levels, we obtain a 250 mil x 250 mil chip. It should be emphasized that the structure of the color bank chip will be facilitated by the development work on the Median Filter.

The switching matrix in the Equivalence Operator is estimated to require less than a 10 mil x 10 mil area for each switching mode leading to a total size of 1000 mils x 1000 mils. The peripheral logic for controlling the matrix

will be two modules 1500 mils x 1500 mils in size. The total area of the Connected Component algorithm is obtained by laying the above elements side by side yielding dimensions, thus far, of 2500 mils x 2500 mils. There are 100 Feature Extractors with a unit size of 12.5 mils x 12.5 mils yielding a area of 1250 mils x 1250 mils. Including the Feature Extractors, the total area of the Connected Components Algorithm is approximately 3750 mils x 3750 mils or 3 3/4 inches x 3 3/4 inches.

The complexity of the Connected Components Algorithm and its size is not insignificant. Development should proceed in steps such that individual components are fabricated separately. A hybrid configuration can be realized by interconnecting all the component chips. With experimental evaluation and testing, progress towards a consolidated version can be realized. The power and yield for the operation and fabrication of the Connected Components Algorithm is difficult to estimate. An alternate approach is to estimate these parameters for the individual components which are comparable in size and complexity to the primitive operators already discussed. Accordingly we feel that the power consumption and yield for each component will be about 100 - 300 milliwatts and approximately 10% respectively.

4.4.2 Focal Plane Area

This section presents a preliminary estimate of the focal plane area occupied by the portions of the cueing system developed thus far, i.e., the image has been smoothed (Median Filter), edges obtained (Gradient Operator), edge width reduced (Non-Maximum Suppression), the image has been segmented into targets (Connected Components), and features extracted (area, height, width, perimeter extent, average gray level). The estimate is preliminary in the sense that none of the clocking circuitry has been included in area estimates for the operators. The reason is that methods by which the algorithms will handle edges of the image frame has not been specified. The estimate does not include the Classifiers.

Assuming that the focal plane is divided into 20 columns, Table 4-1 shows the number of processors required for a system data rate of 1 megapixel/sec. It also shows the geometric area required for each processor and an estimate of the area as defined above. The area thus far is 11¼ inches x 7½ inches. The volume equivalent is 3 inches x 3 inches x 6 inches.

TABLE 4-1. PRELIMINARY ESTIMATE OF FOCAL PLANE AREA

PROCESS	NUMBER REQUIRED	UNIT AREA	TOTAL AREA
Serpentine Delay (36 pixels long)	400	1000 mils x 1 mil for 20	1000 mils x 20 mils
Median Filter	20	100 mils x 128 mils for 1	400 mils x 640 mils
Gradient Operator	20	8 mils x 10 mils for 1	32 mils x 50 mils
Non-Maximum Suppression	20	100 mils x 25 mils for 1	300 mils x 200 mils
Connected Components		3750 mils x 3750 mils for 1	11,250 mils x 7500 mils
			11 1/4 inches x 7 1/2 inches
			3 inches x 3 inches x 6 inches

4.4.3 Chip Development

The Smart Sensor Project is scheduled to last for 21 months with a key circuit selected at the one year mark and constructed in the last nine months. We wanted to select a circuit which was common to as many algorithms as possible.

a. Development Review

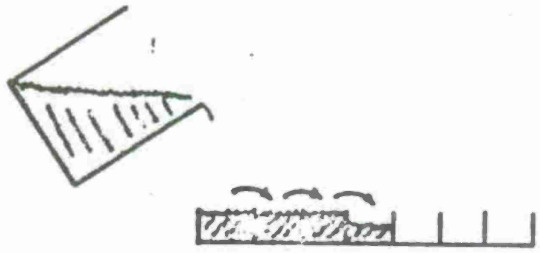
Figure 4.4-1 shows the algorithms developed by Maryland and the functions which are required by each algorithm. A perusal shows that the sorter function occurs in four out of the five algorithms and is the one we selected.

ALGORITHM	FUNCTION
Gradient Operator	Absolute Difference Comparison
Median Filter	Quantizer Sorter
Non Maximum Suppression	Quantizer Sorter Absolute Difference
Connected Components	Quantizer Sorter Coloring Operator
Histogram	Quantizer Sorter

Figure 4.4-1. Algorithm Implementation by CCD Function

Several versions of the sorter were put in production runs at the Westinghouse Advanced Technology Laboratory. Both version assume that the analog signal has already been quantized into a thermometer code. A physical analog is shown in Figure 4.4-2 where a container is filled with an amount of water (charge), proportional to the signal voltage S_1 and then the contents

are poured into a tray of quantized bins. When a bin is filled with water, the water flows over the top into the next bin. The volume of water is divided between a number of discrete bins.



77-0545-V-16

Figure 4.4-2. Flow Analogy to Quantization

One version of the sorter takes the charges, q, residing in each bin and receives them in parallel as in Figure 4.4-3.

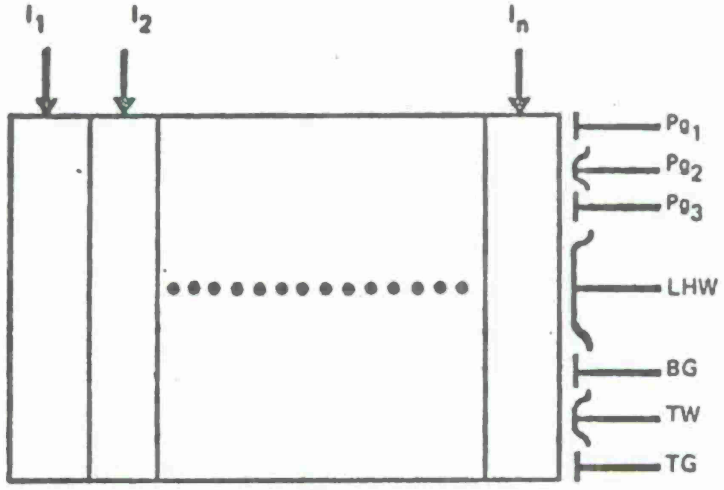


Figure 4.4-3. CCD Sorter

Thus, there is q amount of charge shifted from bin b_1 to channel I_1, q amount of charge shifted from bin b_2 to channel I_2 and so on. By means of gate pg_1, pg_2, and pg_3, the contents of channels I_1 through I_{11} are shifted in parallel into the large holding well LHW. The large holding well is partitioned into

N channels also. Consider a numerical example; a sequence of numbers 4, 7, 5 is quantized at q = 1 so that 4, 7, and 5 bins respectively represent each number. Then Figure 4.4-4 shows the sequence as it goes through the quantizer, the b_1, b_2, ... b_n bins, the I_1, I_2, ... I_n channels and the large holding well. It also shows the removal sequence from the large holding well and the remainder at each stage. The numbers are removed in order of decreasing magnitude 7, 5, 4 which shows the numbers have been sorted by magnitude.

The other version of the sorter is shown in Figure 4.4-5. Here the sorter consists of individually controlled CCD shift registers. Each register is enabled by a clock pulse and the presence of a charge quantum, q. The lower portion of Figure 4.4-5 shows the random sequence 3, 1, 5, 2 of numbers entering the sorter. At stage 2, the first register will shift only. At stage 3, all five registers will shift and note that the data is then arranged in decending order.

We has been using surface channel CCD's to produce image primitives such as produced by the Median Filter and Non-Maximum Suppression Operators because of wider dynamic range and lower data rate requirements. However we found that the Connected Components Algorithm could also be implemented with buried channel CCD devices producing a smaller package so we amended the chip feasibility program to include buried channel devices also. The second version of the sorter was done with surface channel CCD's and the first version was in buried channel. The buried channel device was achieved by ion implantation in the surface channel structure. Probe tests showed that the yields were not high enough to continue processing the buried channel wafers, so they were dropped.

	1	2	3	4	5	6	7	8	9 ..... n
g_i Wells									
5	q	q	q	q	q				
7	q	q	q	q	q	q	q		
4	q	q	q	q					
I_i Channels									
5	q	q	q	q	q				
7	q	q	q	q	q	q	q		
4	q	q	q	q					
Large Holding Well									
4	q	q	q	q					
4,7	2q	2q	2q	2q	q	q	q		
4,7,5	3q	3q	3q	3q	2q	q	q		
First Removal									
	q	q	q	q	q	q	q		
Remainder									
	2q	2q	2q	2q	q				
Second Removal									
	q	q	q	q	q				
Remainder									
	q	q	q	q					
Third Removal									
	q	q	q	q					
Remainder									

Figure 4.4-4. Sorting Sequence

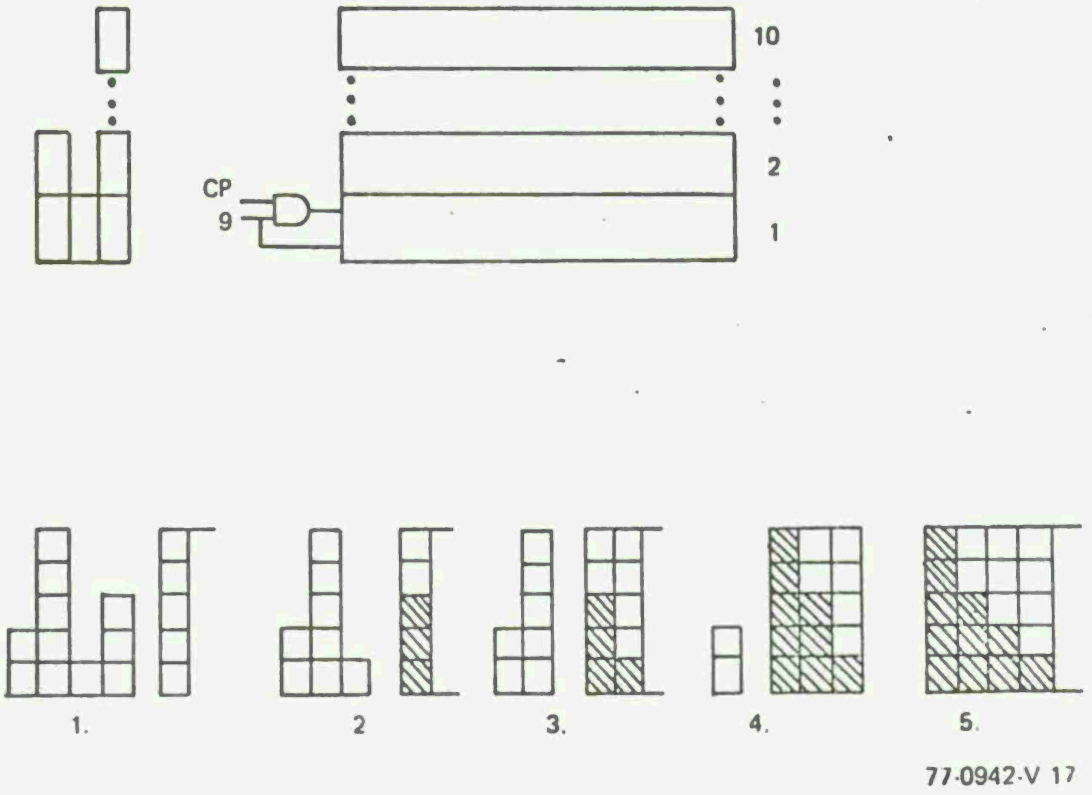


Figure 4.4-5. CCD Sorter Fabrication

The surface channel devices will be used in the demonstration. We could not have met the time schedule if we had started the buried channel devices from the ground up as was done successfully on other programs. For the purposes of this project, amending the lower risk surface channel masks seemed to be a reasonable way to produce both surface and buried channel sorters.

Figure 4.4-6 shows a wafer of the buried channel devices. One portion of the demonstration unit is shown in Figure 4.4-7, with the shift registers mounted in place. The ten shift registers are seen at the top of the unit and ten thumbwheel switches are shown below. These thumbwheels represent the unsorted numbers which the sorter must rearrange in descending order. The observer may dial in any arrangement of numbers which he wishes. The outputs

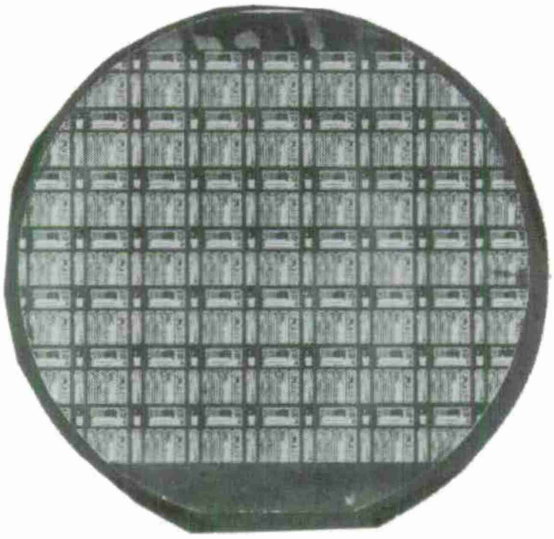


Figure 4.4-6. Buried Channel Wafer

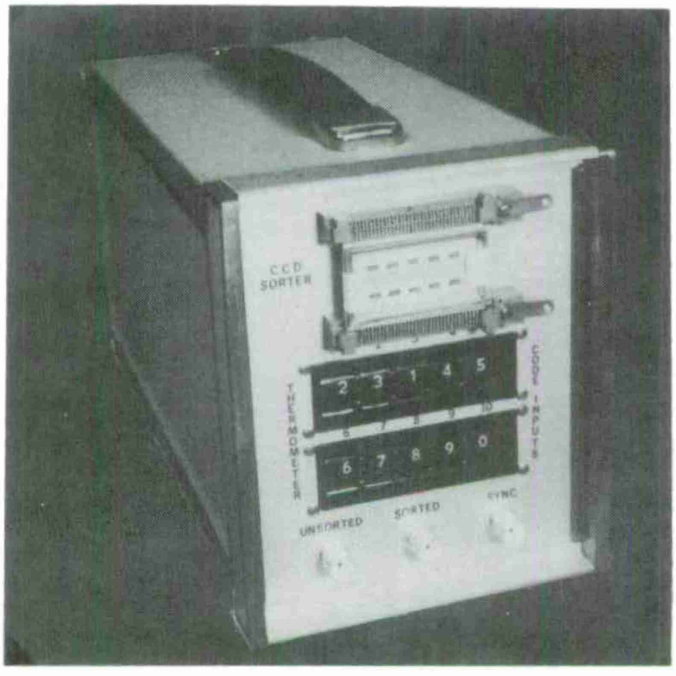


Figure 4.4-7. Demonstration Unit

and inputs, i.e., the unsorted and sorted arrangements are shown on a two-trace oscilloscope.

Westinghouse IR&D accounted for 70 cents of every dollar spent on the Smart Sensor Project.

b. Feasibility Demonstration

The demonstration unit and a two-trace oscilloscope were exhibited at the DARPA Image Understanding Symposium in September of 1977 at Stanford University in Palo Alto, California. The symposium participants were encouraged to dial in their own set of random numbers on the thumbwheel switches and observe the ordered outputs. Figure 4.4a, b, c, and d are typical outputs of the two-trace oscilloscope seen at the Symposium. The random sequence is shown in the left half of the trace and the ordered sequence in the right half.

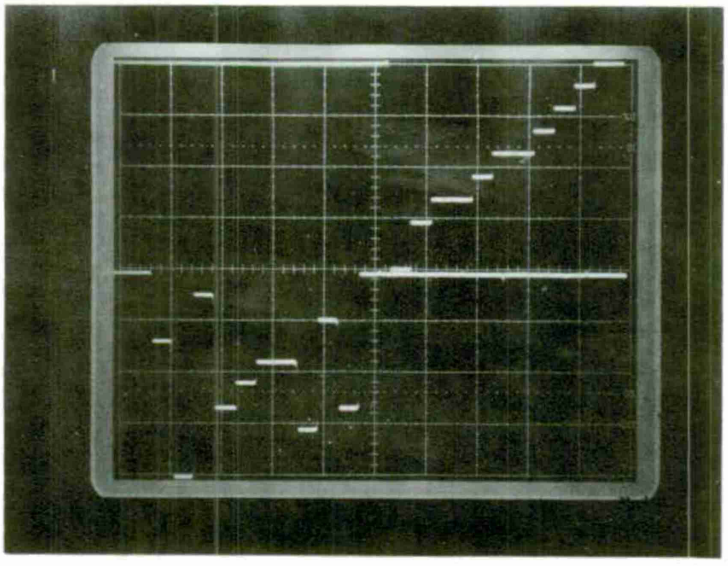


Figure 4.4a. Sorter Display

Figure 4.4a, is a sequence of 6, 0, 8, 3, 4, 5, 5, 2, 7, 3 arranged into an ordered sequence of 0, 2, 3, 3, 4, 5, 5, 6, 7, 8. Figure 4.4b is a sequence of 4, 4, 4, 4, 4, 5, 4, 4, 4, 4 arranged in an ordered sequence of 4, 4, 4, 4, 4, 4, 4, 4, 4, 5. Figure 4.4c is a sequence of 4, 7, 4, 5, 4, 1, 4, 4, 3, 4 arranged in an ordered sequence of 1, 3, 4, 4, 4, 4, 4, 4, 5, 7. Figure 4.4d is a sequence of 4, 4, 4, 4, 4, 3, 4, 4, 4, 4 arranged in an ordered sequence of 3, 4, 4, 4, 4, 4, 4, 4, 4, 4. The unit was also demonstrated at the Night Vision Laboratory, Ft. Belvoir, Va. on November 28, 1977.

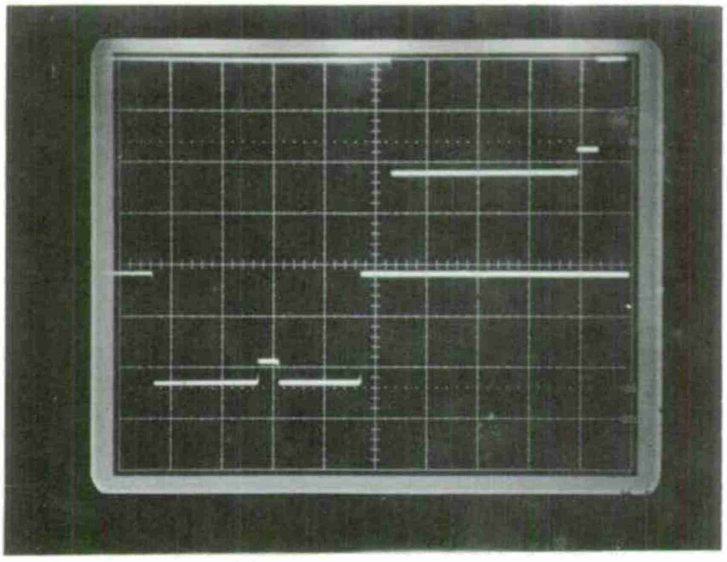


Figure 4.4b. Sorter Display

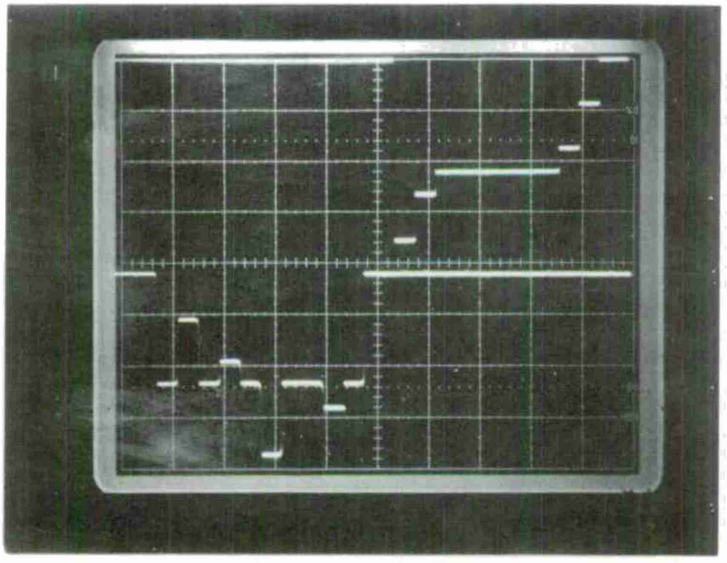


Figure 4.4c. Sorter Display

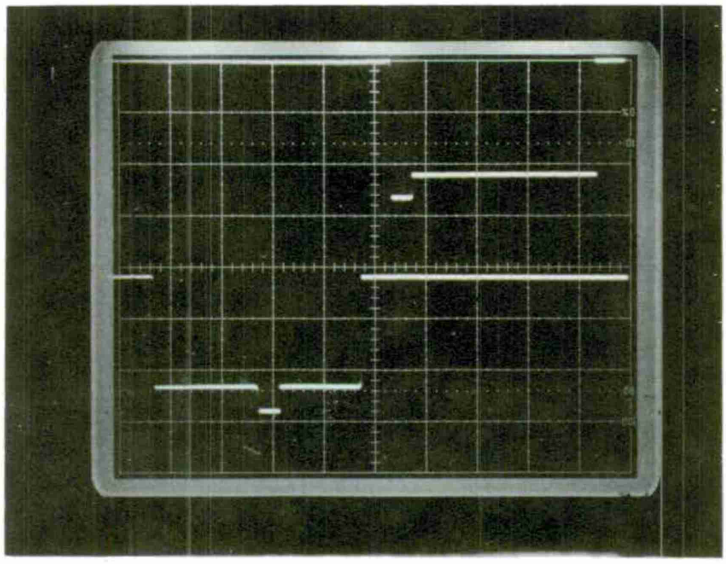


Figure 4.4d. Sorter Display

4.5 Conclusions and Recommendations

This report has shown that the Smart Sensor can be implemented with CCD technology and in a smaller package than implementation with digital techniques. However, it is also evident that the CCD package package cannot be accomodated on the focal plane.

The total area estimate is 11 1/4 inches by 7 1/2 inches . If 3 inch by 3inch modules were employed with 1/2 inch centers, the volume dimensions would be 3 inches by 3 inches by 6 inches. This volume has positive implications for missiles and other airborne platforms, in general.

The next step in proving feasibility involves building some of the modules and checking them for numerical integrity, size, speed, and power consumption. These modules, e.g. Median Filter would be hybrid packages in the first build, and the clocking circuitry should be included on the chip. Another item of particular interest is the Connected Components Algorithm with the switching matrix and peripheral control logic. Estimates of ultimate size for the monolithic elements would be necessary as well as estimates on the grouping of algorithms within the monolthic packages.

5. Publications

In accordance with contract provisions, the University of Maryland produced seven Quarterly Technical Management Reports, a First Quarter Report, and three Semiannual Reports. Westinghouse, as subcontractor, prepared six Quarterly Technical Progress Reports. In addition, the following sixteen Computer Science Technical Reports, twelve Image Understanding Workshop papers and five other proceedings papers were produced with the support of this contract.

a) Computer Science Technical Report Series

Durga P. Panda, "A Method of Adaptive Smoothing and Edge Enhancement", TR-504, February 1977.

Durga P. Panda, "Segmentation of FLIR Images by Pixel Classification", TR-507, February 1977.

David L. Milgram, "Region Tracking Using Dynamic Programming", TR-539, May 1977.

David L. Milgram, "Constructing Trees for Region Description", TR-541, May 1977.

Durga P. Panda, "Statistical Analysis of Some Edge Operators", TR-558, July 1977.

Durga P. Panda, "Statistical Properties of Thresholded Images", TR-559, July 1977.

Robert A. Hummel, "Edge Detection Using Basis Functions", TR-569, August 1977.

Yasuo Nakagawa and Azriel Rosenfeld, "A Note on the Use of Local MIN and MAX Operations in Digital Picture Processing", TR-590, October, 1977.

Shmuel Peleg,"Iterative Histogram Modification, 2", TR-606, November 1977.

Charles R. Dyer and Azriel Rosenfeld, "Thinning Algorithms for Grayscale Pictures", TR-610, November 1977.

Kenneth C. Hayes, Jr., Martin Herman, and Russell Smith, "PDP-11 Image Processing Software", TR-612, December 1977.

Gilbert B. Shaw, "Local and Regional Edge Detectors: Some Comparisons", TR-614, December 1977.

David L. Milgram and Martin Herman, "Clustering Edge Values for Threshold Selection", TR-617, December 1977.

David L. Milgram and Daryl J. Kahl, "Recursive Region Extraction", TR-620, December 1977.

Yasuo Nakagawa and Azriel Rosenfeld, "Some Experiments in Variable Thrsholding", TR-626, January 1978.

Martin Herman, "A System for Control Structure Implementation for Image Understanding", TR-646, March 1978.

b) DARPA Image Understanding Workshops: Proceedings

1. April 20, 1977. Minneapolis, Minnesota.

David L. Milgram, "Region Extraction Using Convergent Evidence", pp. 58-64

Durga P. Panda, "Segmentation of FLIR Images by Pixel Classification", pp. 65-70

Thomas Willett and Nathan Bluzer, "Automatic Target Cueing on the Focal Plane", pp. 87-88

Azriel Rosenfeld, "Algorithms and Hardware Technology for Image Recognition -- Project Status Report - March, 1977," pp. 98-100.

2. October 20,21, 1977. Palo Alto, California

Azriel Rosenfeld, "Relaxation Methods - Recent Developments", pp. 28-30.

T. Schutt, G. Borsuk and T. J. Willett, "A CCD Histogram-Sorter: Feasibility Chip", pp. 7-8.

David L. Milgram, "Progress Report on Segmentation Using Convergent Evidence", pp. 104-108.

David L. Milgram, Azriel Rosenfeld, "Algorithms and Hardware Technology for Image Recognition, Project Status Report-September 1977", pp. 139-140.

3. May 3-4, 1978. Cambridge, Massachusetts

T. J. Willett, G. E. Tisdale, "Hardware Implementation of a Smart Sensor: A Review", pp. 1-8.

Azriel Rosenfeld, "Some Recent Results Using Relaxation-Like Processes", pp. 100-104.

David L. Milgram, "Results in FLIR Target Detection and Classification" , pp. 118-124.

Azriel Rosenfeld, David L. Milgram, "Algorithms and Hardware Technology for Image Recognition", p. 142.

c) Other Publications

Thomas Willett, Nathan Bluzer, "Automatic Target Cueing on the Focal Plane", Proceedings of the Seventh Annual Symposium on Automatic Imagery Pattern Recognition, May 23-24, 1977, College Park, MD, pp. 8-9.

David L. Milgram, "Automatic Object Detection in FLIR Images", Proccedings of the International Conference on Cybernetics and Society, September 19-21, 1977, Washington, D.C., pp. 617-621.

Durga P. Panda, "A Method of Adaptive Smoothing and Edge Enhancement", Proceedings of the International Conference on Cybernetics and Society, September 19-21, 1977, Washington, D.C., pp. 648-656.

Charles R. Dyer, David L. Milgram, "VIEWMASTER - A System for Building Image Processing Programs", Proceedings of the Eighth Annual Symposium on Automatic Imagery Pattern Recognition, April 3-4, 1978, Gaithersburg, MD, pp. 170-179.

6. Distribution List

NIGHT VISION LABORATORY
DISTRIBUTION LIST FOR TECHNICAL REPORTS

* 1 Copy Unless otherwise specified

Defense Documentation Center
ATTN: DDC-TCA
Cameron Station (Bldg 5)
Alexandria, VA 22314

*12 cys unlimited distribution
2 cys limited or classified

Director
National Security Agency
ATTN: TDL
Fort George G. Meade, MD 20755

Office of Naval Research
Code 427
Arlington, VA 22217

Director
Naval Research Laboratory
ATTN: Code 2627
Washington, DC 20375

Commander
Naval Electronics Laboratory Center
ATTN: Library
San Diego, CA 92152

Commander
US Naval Ordnance Laboratory
ATTN: Technical Library
White Oak, Silver Spring, MD 20910

Commandant, Marine Corps
HQ, US Marine Corps
ATTN: LMC
Washington, DC 20380

HQ, US Marine Corps
ATTN: Code INTS
Washington, DC 20380

Command, Control & Communications Div.
Development Center
Marine Corps Dev. & Edu Cmd
Quantico, VA 22134

HQ, ESD (XRRI)
L. G. Hanscom Field
Bedford, MA 01730

Air Force Avionics Laboratory
ATTN: AFAL/TSR, STINFO
Wrignt-Patterson AFB OH 45433 *2

AFSPCOMMCEN/SUR
San Antonio, TX 78243

Armament Development & Test Center
ATTN: DLOSL, Tech Library
Eglin Air Force Base, FL 32542

HQ DA (DACE-CMS)
Washington, DC 20310

OSASS-RD
Washington, DC 20310

Commander
US Army Training & Doctrine Command
ATTN: ATCD-SI
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCDOCI
Fort Monroe, VA 23651

Commander
US Development & Readiness Command
ATTN: DRCMA-EE
5001 Eisenhower Ave
Alexandria, VA 22333

Commander
US Army Development & Readiness Command
ATTN: DRCRD-FW
5001 Eisenhower Ave.
Alexandria, VA 22333

Commander
US Army Training & Doctrine Command
ATTN: ATCD-F
Fort Monroe, VA 23651

Commander
US Army Missile Command
ATTN: DRSMI-RR, Dr. J.P. Hallowes
Redstone Arsenal, AL 35809

Commander
US Army Armament Command
ATTN: DRSAR-RDP(Library)
Rock Island, IL 61201

Commander
US Army Combined Arms Combat
Developments Activity
ATTN: ATCAIC-IE
Fort Leavenworth, KS 66027 *3

Commander
US Army Logistics Center
ATTN: ATCL-MA
Fort Lee, VA 23801

Commandant
US Army Ordnance School
ATTN: ATSOR-CTD
Aberdeen Proving Ground, MD 21005

Commander
US Army Intelligence School
ATTN: ATSIT-CTD
Fort Sill, OK 73503

Commandant
US Army Engineer School
ATTN: ATSE-CTD-DT-TL
Fort Belvoir, VA 22060

Commander
Picatinny Arsenal
ATTN: SARPA-TS-S #59
Dover, NJ 07801

Commander
Frankford Arsenal
ATTN: (Dr. Wm. McNeill) PDS
Philadelphia, PA 19137

Commander
USASA Test & Evaluation Center
Fort Huachuca, AZ 85613

US Army Research Office - Durham
ATTN: CRDARD-IP
Box CM, Duke Station
Durham, NC 27706

Commander
HQ MASSTER
Technical Information Center
ATTN: Mrs. Ruth Reynolds
Fort Hood, TX 76544

USA Security Agency
ATTN: IARD
Arlington Hall Station
Arlington, VA 22212

Commander
US Army Tank Automotive Command
ATTN: DRSTA-RW-L
Warren, MI 48090

Commandant
US Army Air Defense School
ATTN: C&S Dept, Msl Sci Div
Fort Bliss, TX 79906

Commander
US Army Combined Arms Combat
Developments Activity
ATTN: ATCACC
Fort Leavenworth, KS 66027

Commander
US Army Yuma Proving Ground
ATTN: STEYP-MTD (Tech Library)
Yuma, AZ 85364 *2

Commander
US Army Arctic Test Center
ATTN: STEAC-PL
APO Seattle, WA 98733

Commander
US Army Tropic Test Center
ATTN: STETC-MO-A (Tech Library)
Drawer 942
Ft. Clayton, Canal Zone 09827

Commander
US Army Logistics Center
ATTN: ATCL-MC
Ft. Lee, VA 22801

Directorate of Combat Developments
US Army Armor School
ATTN: ATSB-CD-AA
Ft. Knox, KY 40121

Commandant
US Army Inst for Military Assistance
ATTN: ATSU-CTD-OMS
Ft. Bragg, NC 28307

Commander
US Army Missile Command
ATTN: DRSMI-RE (Mr. Pittman)
Redstone Arsenal, AL 35809

Commander
US Army Systems Analysis Agency
ATTN: DRXSY-T (Mr. A. Reid)
Aberdeen Proving Ground, MD 21005

Commandant
US Army Signal School
ATTN: ATSN-CTD-MS
Ft. Gordon, GA 30905

Commander
US Army Tank Automotive Command
ATTN: DRSTA-RHP (Dr. J. Parks)
Warren, MI 48090

NASA Scientific & Tech Info Facility
ATTN: Acquisitions Br. (S-AK/DL)
PO Box 33
College Park, MD 20740 *2

Advisory Group on Electron Devices
201 Varick St, 9th Floor
New York, NY 10014

Ballistic Missile Radiation Anal Ctr
Env. Research Insti. of Michigan
Box 618
Ann Arbor, MI 48107

Chief
Ofc of Missile Electronic Warfare
Electronic Warfare Lab, ECOM
White Sands Millile Range, NM 88002 *2

Chief
Intel Materiel Div & Support Ofc
Electronic Warfare Lab, ECOM
Ft. Meade, MD 20755

Commander
UA Army Electronics Command
ATTN: DRSEL-MS-TI
Fort Monmouth, NJ 07703

TACTEC
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Commander
US Army Electronics Command
ATTN: DRSEL-PL-ST
Fort Monmouth, NJ 07703

Study Center
National Maritime Research Center
ATTN: Rayma Feldman
King's Point, NY 11024

IR Information & Analysis Ctr
PO Box 618
ATTN: Library
Ann Arbor, MI 78107

Advanced Research Projects Agency
ATTN: IPTO (Maj. Carlstrom)
1400 Wilson Blvd.
Arlington, VA 22209 *4

Commanding General
US Army Development & Readiness Command
Attn: DRXSP (B. Chasnov)
5001 Eisenhower Ave
Alexandria, VA 22333

Commanding Officer
Naval Underwater Center
ATTN: Code 6003 (H. Whitehouse)
San Diego, CA

Commanding Officer
US Air Force Avionics Lab
Bldg 620, ATTN: AFAL/AAT (Dr. N. Griswold)
Wright-Patterson AFB, Ohio 45433

US Army Research Office
ATTN: Dr. J. Suttle
Box CM, Duke Station
Durham, NC 27706

Director
Night Vision Laboratory
ATTN: DRSEL-NV-VI (Dehne)
Ft. Belvoir, VA 22060 *27/1

Director
Night Vision Laboratory
ATTN: DRSEL-NV-FIR (Kelso)
Ft. Belvoir, VA 22060 *15

Director
Night Vision Laboratory
ATTN: DRSEL-NV-D (Fulton)
Ft. Belvoir, VA 22060 *12

Dr. Ian Henderson
Defense Research Establishment Ottawa
National Defense H.Q.
Ottawa, Canada

Dr. Lewis Minor
U.S. Army Missile Research and
Development Command
DRDMI-TEO
Redstone Arsenal
Huntsville, Alabama 35809

Dr. Claude Roche
Establissement Technique Central
de L'Armement
16 bis, Avenue Prieur de la
Cote d'Or.
94114 Arcueil, France

Mr. Lars Brock-Nannestad
Danish Defense Research Establishment
Osterbrogades Kaserne
DK-2100 Copenhagen, O, Denmark

Dr. Manfred Bohner
FIM/FGAN
Breslauer Strasse 48
7500 Karlsruhe 1
Federal Republic of Germany

Mr. Fons M. Navarro
Physics Laboratory TNO
Oude Waalsdorperweg 63
The Hague, Netherlands

Mr. Stein Grinaker
Norwegian Defense Research
Establishment
Division for Weapons and Equipment
PBox 25
N-2007 Kjeller, Norway

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ALGORITHMS AND HARDWARE TECHNOLOGY FOR IMAGE RECOGNITION		5. TYPE OF REPORT & PERIOD COVERED Final 1 May 1976-31 Jan. 1978 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David L. Milgram, Thomas J. Willett Azriel Rosenfeld, Glenn E. Tisdale (Univ. of Md.) (Westinghouse Corp.)		8. CONTRACT OR GRANT NUMBER(s) DAAG53-76-C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Center University of Maryland College Park, MD 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Night Vision Laboratory Fort Belvoir, VA 22060		12. REPORT DATE 31 March 1978 13. NUMBER OF PAGES 291
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Image understanding
Pattern recognition
Target detection
FLIR imagery

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Algorithms for detecting and classifying targets on forward-looking infrared (FLIR) imagery have been developed. The implementability of these algorithms in charge-coupled device (CCD) technology has been investigated, and one basic function, sorting, has been successfully implemented. The recommended algorithms had a detection rate above 95% and a false alarm rate of 1 to 2 per frame.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*